ROMÂNIA
MINISTERUL APĂRĂRII NAŢIONALE
ACADEMIA TEHNICĂ MILITARĂ
“FERDINAND I”

FACULTATEA DE SISTEME INFORMATICE ŞI SECURITATE CIBERNETICĂ
Specializarea: Securitatea Tehnologiei Informației

Serviciu de validare a semnăturilor digitale în conformitate cu standardele ETSI

COORDONATOR ŞTIINŢIFIC:
DE COMPLETAT dr. ing. Mihai-Lică PURA

MASTERAND:
Slt. Ing. Ștefan BODOARCĂ

Conţine _____ file
Inventariat sub nr_____
Poziţia din indicator: ____
Termen de păstrare: _____

BUCUREŞTI
2020

ABSTRACT

Enormous amounts of electronic transactions are everyday used in many traditional and newer businesses. Many applications are nowadays carrying out sensible and personal data, which have led to a critical need for protecting the information, for ensuring the authenticity and for suporting nonrepudiation. Just as signatures facilitate validation and verification of the authenticity of paper documents, digital signatures are used for the validation and authentication of electronic docments.

This paper presents a signature validation service developed in order to meet the requirements described in the ETSI technical specifications for the validation of digital signatures.

The theoretical aspect of this paper, meaning chapters one to four, gives an overview on digital signatures with accent on their formats of representation, and also presents the signature validation process according to ETSI standards.

Chapter five presents the developed system for the vadalition of digital signatures and digital certificates.

The last chapter presents the conclusions of this paper.

REZUMAT

În contextul societății actuale tranzacțiile electronice sunt folosite în aproape toate domeniile de activitate. Astfel, foarte multe aplicații gestionează date sensibile și personale, ceea ce a condus la o nevoie stringentă în ceea ce privește protejarea informației, asigurarea integrității informației și a non-repudierii sursei de informație. La fel cum semnăturile olografe sunt folosite pentru documentele fizice, semnăturile electronice sunt folosite pentru documentele electronice.

Această lucrare prezintă un serviciu de validare a semnăturilor dezvoltat cu scopul de a răspunde specificațiilr tehnice definite de ETSI cu privire la validarea semnăturilor digitale.

Aspectele teoretice ale tezei, însemnând capitolele de la unu la patru, cuprind prezentarea semnăturilor digitale punând accent pe formatele ETSI de reprezentare a semnăturilor și prezintă la nivel tehnic procesul de validare a unei semnături digitale.

Capitolul 5 prezintă sistemul dezvoltat pentru validarea semnăturilor și a certificatelor digitale.

Ultimul capitol adresează concluziile lucrării.

Cuprins

Capitolul 1

Introducere

Structura societății actuale este, în mod incontestabil, influențată de rețelele de comunicații și în special de Internet. Calculatoarele sunt utilizate pe scară largă în tot mai multe sectoare de activitate: comerț, efectuare de plăți on-line, vot, sectoare ale apărării și securității naționale, etc. Din nefericire, beneficiile oferite de Internet vin la pachet cu riscuri și amenințări în ceea ce privește securitatea informației. Astfel, informații personale (parole, detaliile unei plăți on-line) pot fi interceptate pe liniile de comunicații, date critice (licitație on-line, vot electronic) pot fi manipulate în mod malițios dacă tranzacțiile electronice nu sunt protejate corespunzător. În acest context, rezultă că nu orice tranzacție on-line poate fi considerată de încredere, că nu orice identitate afirmată în mediul on-line este și cea reală, etc.

În viața de zi cu zi, tranzacțiile dintre persoane/entități pot fi sigilate prin semnătură olografă - documentele aferente tranzacției sunt semnate. Plecând de la modelul semnăturilor olografe s-au construit semnăturile electronice. O comparație între aceste două tipuri de semnături este realizată la 3.1. Semnăturile electronice sunt folosite pentru semnarea documentelor în format electronic, după cum semnăturile olografe sunt folosite pentru documentele fizice. Semnăturile electronice pot suferi atacuri malițioase fiind până la urmă date ce ajung să circule prin rețeaua de comunicație. În consecință, trebuie să existe metode de verificare a acestor semnături.

Lucrarea de față are în centru prezentarea unui serviciu de validare a semnăturilor digitale conform standardelor ETSI. Institutul European pentru Standarde în Telecomunicații - *European Telecommunications Standards Insititute* - (ETSI) este o organizație independentă, nonprofit, de standardizare din industria telecomunicațiilor (producători de echipamente și operatori de rețea) din Europa, cu sediul în Sophia-Antipolis, Franța, cu proiecție la nivel mondial. ETSI produce standarde globale aplicabile tehnologiilor informației și comunicațiilor (TIC), inclusiv tehnologiilor fixe, mobile, radio, convergente, broadcast și tehnologiilor internet [35]. ETSI publică anual între 2.000 și 2.500 de standarde. De la înființarea sa în 1988, a produs peste 30.000. Acestea includ standarde care adresează tehnologii globale foarte importante, cum ar fi sistemul de telefonie mobilă GSM, 3G, 4G, DECT, sistemul mobil radio TETRA, dispozitive de distanță scurtă, incluzând radio LPD, carduri inteligente și multe altele [35]. În lucrarea de față se iau în considerare standardele ETSI privitoare la validarea semnăturilor digitale.

Lucrarea este structurată pe 6 capitole.

Capitolul 1, mai exact capitolul curent, prezintă necesitatea dezvoltării unui mecanism de verficare a semnăturilor digitale în conformitate cu standardele ETSI, oferă câteva informații cu privire la ETSI și prezintă structura lucrării pe capitole.

Capitolul 2 enumeră abrevierile folosite în teza de față și definește o serie de termeni necesari înțelegerii anumitor aspecte prezentate în memoriul tehnic. Acești termeni au fost definiți în capitolul al doilea, deoarece nu reprezintă, în mod direct, obiectul discuției de la parte teoretică.

Capitolul 3 introduce semnăturile digitale, descrie un model de creare și verificare a acestora, enumeră entitățile implicate într-o tranzacție electronică și prezintă formatele semnăturilor digitale. Prezentarea formatelor se axează în principal pe formatele de smnătură electornică avansată definite de ETSI.

Capitolul 4 prezintă procesul de validare a semnăturii digitale definit de specificațiile tehnice realizate de ETSI. În acest context, se prezintă părțile implicate într-un proces de validare a semnăturii, arhitectura unui serviciu de validare, model conceptual și pașii parcurși pentru validare, rezultatele procesului de validare, politicile de validare și procesele propriu-zise de validare a semnăturii.

Capitolul 5 prezintă sistemul implementat pentru validarea semnăturilor și a certificatelor digitale. Tot aici se prezintă funcționalitățile seriviciului de validare. De subliniat este faptul că sistemul implementat are scop validarea semnăturii, dar și validarea certificatului pentru a putea fi folosit într-un context intern al unei organizații (într-o rețea internă), fără necesitatea extragerii anumitor informații, în mod obligatoriu, din Internet. Capitolul 5 prezintă tehnologiile folosite, componentele sistemului și interacțiunea dintre capitole și se încheie cu testarea și validarea sistemului.

Capitolul 6 prezintă concluziile lucrării.

Capitolul 2

Abrevieri și Definiții

2.1 Abrevieri

ETSI	European Telecommunications Standards Institute
TS	Technical Specification
CMS	Cryptographic Message Syntax
CAdES	CMS Advanced Electronic Signature
CAdES-BES	CAdES Basic Electronic Signature
CAdES-C	CAdES with Complete validation data
CAdES-EPES	CAdES Explicit Policy Electronic Signature
CAdES-LT	CAdES Long Term
CAdES-T	CAdES with Time
CAdES-X Long	CAdES with Extended Long validation data
CAdES-X	CAdES with Extended validation data
CAdES-XL	CAdES-X Long
SHA	Secure hash algorithm
CEN	European Committee for Standardization
PKCS	Public Key Cryptography Standards
RSA	Ron Rivest, Adi Shamir, Leonard Adleman
RFC	Request For Comments
MAC	Message Authentication Code
OID	Object Identifier
PDF	Portable Document Format
XML	Extensible Markup Language
PAdES	PDF Advanced Electronic Signature
XAdES	XML Advanced Electronic Signature

XAdES-BES	XAdES Basic Electronic Signature
XAdES-T	XAdES with Time
XAdES-C	XAdES with Complete validation data
XAdES-X	XAdES with Extended validation data
XAdES-XL	XAdES-X Long
XAdES-A	XAdES Archiving validation data
CAdES-A	CAdES Archiving validation data
W3C	World Wide Web Consortium
URI	Uniform Resource Identifier
ESI	Electronic Signatures and Infrastructures
TSP	Trust Service Provider
SVSP	Signature Validation Service Provider
CA	Certification Authority
TSA	Time Stamping Authority
SVA	Signature Validation Application
DA	Driving Application
(Q)SCD	(Qualified) Signature Creation Device
LOTL	List of Trusted Lists sau List of the Lists
SVSServ	Signature Validation Service Server
SVP	Signature Validation Protocol
OCSP	Online Certificate Status Protocol
CRL	Certificate Revocation List
LT	Long Term
LTV	Long Term Validation
POE	Proof of existence
SAV	Signature Acceptance Validation
CV	Cryptographic Verification
XCV	X.509 Certificate Validation

eIDAS	Electronic Identification, Authentication and Trust Services
DSS	Digital Signature Service
API	Application Programming Interface
REST	Representational state transfer
HTTP	HyperText Transfer Protocol
HTTPS	HTTP/Secure
AIA	Authority Information Access
TLS	Transport Layer Security
SSL	Secure Sockets Layer
SQL	Structured Query Language
DB	Database
HSQLDB	HyperSQL Database
JDBC	Java Database Connectivity
RAM	Random Access Memory
IMDB	In-memory Database
EU	European Union
PEM	Privacy-Enhanced Mail
IETF	Internet Engineering Task Force
ASiC	Associated Signature Containers
ASiC-S	ASiC-Simple
ASiC-E	ASiC-Extended
LPD	Low Power Device
PSC	Point of Single Contact

În teza de față, se definesc următoarele abrevieri care nu sunt un standard:

SignatureLTVM	Signature with Long-Term Validation Material
SignatureLTAIVM	Signature providing Long Term Availability and Integrity of Validation Material

2.2 Definiții

1. **Basic Signature** - semnătură de bază - este o semnătură care poate fi validată atât timp cât certificatul nu este revocat sau expirat (Anexa 1 Figura 2). [12]
2. **Signature with Time** - semnătură cu marcă temporală - dovedește că semnătura a existat la un anumit moment de timp (Anexa 1 Figura 3). [12]
3. **Signature with Long-Term Validation Material** - semnătură care furnizează disponibilitatea pe termen lung a datelor folosite pentru validare, incluzând toate materialele sau referințe către materialele necesare pentru validarea semnăturii (Anexa 1 Figura 4). [12]
4. **Signature providing Long Term Availability and Integrity of Validation Material** - vizează disponibilitatea pe termen lung și integritatea datelor folosite la validarea semnăturilor pe termen lung și pot oferi sprijin pentru verificarea unei semnături dincolo de evenimente ce ar putea limita validitatea (de exemplu, învechirea/slăbirea algoritmilor criptografici, expirarea datelor de validare) (Anexa 1 Figura 5). [12]
5. **Semnătură digitală înfășurată** - *eng. enveloped (digital) signature* - semnătură digitală încorporată în obiectul de date semnat. [12]
6. **Semnătură digitală înfășurătoare** - *eng. enveloping (digital) signature* - semnătură digitală care încorporează obiectul de date semnat. [12]
7. **Semnătură detașată** - *eng. detached (digital) signature* - semnătură care, cu referire la obiectul de date, nu este nici înfășurată nici înfășurătoare [12]; este o semnătură separată de obiectul semnat.
8. **Acceptarea semnăturii -** *eng. Signature acceptance* - verificare tehnică care trebuie efectuată pe semnătura în sine sau pe elementele semnăturii. [12]
9. **Clasă de semnături** - set de semnături ce îndeplinesc o anumită funcționalitate [12]. Elementele definite anterior (1. , 2. , 3. , 4.) reprezintă clase de semnături.
10. **Puncte unice de contact** - *eng. Points of Single Contact* - portaluri de *e-goverment* care permit furnizorilor de servicii să obțină informațiile de care au nevoie și să completeze procedurile administrative on-line. [36]

Capitolul 3

Semnături digitale

3.1 Caracterisiticile semnăturilor scrise și ale semnăturilor digitale

O semnătură obișnuită trebuie să aibă următoarele caracteristici: stabilirea cu ușurință a autenticității semnăturii, dificultatea falsificării semnăturii, semnătura să nu poată fi transferată altei persoane, non-repudierea astfel încât semnatarul să nu poată nega propria semnătură. [1]

O semnătură digitală trebuie să aibă toate caracteristicile amintite pentru o semnătură obișnuită, plus câteva caracteristici specifice, deoarece acest tip de semnare este folosit în mod practic în aplicații ce au caracter sensibil precum schimbul sigur de e-mail-uri, tranzacții bancare peste Internet, etc. Din moment ce o semnătură digitală este doar o secvență de 0 și 1, este de dorit ca aceasta să aibă următoarele proprietăți: semnătura să fie o înșiruire de biți care să depindă de mesajul semnat (astfel, semnătura va fi diferită pentru documente diferite, chiar dacă semnatarul este același); semnătura trebuie să folosească informație care este unică pentru semnatar pentru a preveni falsificarea și negarea semnării; trebuie să fie relativ ușor de produs; trebuie să fie relativ ușor de recunoscut, iar autenticitatea acesteia să fie ușor de verificat. Din punct de vedere computațional, o semnătură digitală trebuie să fie imposibil de falsificat, ceea ce înseamnă că nu se poate construi un mesaj nou pentru o semnătura deja existentă sau că nu se poate construi o semnătură falsă pentru un anumit mesaj. De asemenea, trebuie ca semnăturile digitale să poată fi păstrate pe mecanisme de stocare pentru a rezolva posibile dispute mai târziu. [1]

Pentru a verifica autenticitatea emițătorului unui document și a faptului că acest document nu a fost modificat, au fost dezvoltate câteva proceduri, numite tehnici de autentificare. Totuși, tehnicile de autentificare a mesajelor nu pot fi folosite direct ca semnături digitale, deoarece au anumite neajunsuri. Spre exemplu, cu toate că autentificarea mesajului protejează cele două părți implicate în schimbul de mesaje de o terță parte, nu protejează cele două părți una de cealaltă. Mai mult, schemele elementare de autentificare produc semnături care au aceeași lungime ca mesajele în sine. [1]

3.2 Noțiuni de bază și de terminologie

Semnăturile digitale sunt compuse pe baza documentelor (mesaje/informație) care trebuie semnate și pe baza unor informații private,

deținute numai de semnatar. În practică, în loc să se folosească tot mesajul în procesul de semnare digitală, se aplică o funcție de hash peste mesaj pentru a se obține un rezumat al acestuia (eng. *message digest*). O funcție de hash primește la intrare un mesaj de lungime arbitrară și produce un rezumat de lungime fixă al mesajului. Funcții de hash precum SHA-1, SHA-2, SHA-3 asigură faptul că este foarte puțin probabil ca din două mesaje diferite să se obțină același hash. Există două largi tehnici ce pot fi folosite în formarea unei semnături digitale – criptosisteme cu chei simterice și criptosisteme cu chei publice (criptosistem, în sens larg, se referă la o tehnică de criptare). În criptosistemul cu chei simetrice este folosită o cheie secretă cunoscută doar de emițător și de receptorul legitim al mesajului. Cu toate acestea, trebuie să existe o cheie unică între oricare doi utilizatori. Astfel, odată cu creșterea numărului de utilizatori devine extrem de dificilă generarea, distribuirea și ținerea în evidență a acestor chei secrete. [1] Desigur, aceste criptosisteme, atunci când sunt folosite pentru generarea unei semnături digitale, prezintă aproximativ aceleași neajunsuri ca tehnicile de autentificare a mesajelor, diferența constând în faptul că aceste semnături pot avea dimensiune variabilă. În practică, pentru generarea de semnături digitale se folosesc criptosistemele cu chei publice.

Un criptosistem cu chei publice folosește o pereche de chei: o cheie privată, cunoscută doar de către cel care o deține și o cheie publică, cunoscută de oricine dorește să comunice cu respectiva persoană. Pentru ca mesajul trimis să fie confidențial acesta este criptat cu cheia publică a receptorului, putând fi acum decriptat doar de receptor cu cheia privată asociată cheii publice. În ceea ce privește autentificarea, un mesaj poate fi criptat cu cheia privată a emițătorului, pe care îl vom referi ca A. Acest mesaj poate fi acum decriptat de oricine folosește cheia publică a lui A. Dacă în urma decriptării se obține mesajul inițial, atunci este evident că mesajul a fost criptat de cheia privată a lui A, astfel că doar A ar fi putut să trimită mesajul. [1]

3.3 Crearea și verificarea unei semnături digitale

O schemă simplă de creare și verificare a unei semnături digitale este prezentată în Figurile 1 și 2. O funcție de hash este aplicată peste mesaj, obținându-se un rezumat al mesajului de dimensiune fixă. Funcția de semnare folosește hash-ul mesajului și cheia privată a emițătorului pentru a genera semnătura digitală. O formă foarte simplă de semnătură digitală este obținută prin criptarea hash-ului mesajului cu cheia privată a emițătorului. Mesajul nu este criptat și poate fi citit de oricine. Cu toate acestea, semnătura asigură autenticitatea emițătorului. Mesajul în clar împreună cu semnătura poate fi comparat cu un document publicat de o autoritate oficială, acest document conținând și o semnătură care să ateste autenticitatea mesajului. Pentru receptor,

este aplicată funcția inversă, practic decriptarea, obținându-se astfel hash-ul mesajului. Mesajul primit în clar este trecut acum prin aceeași funcție de hash ca mesajul original. Acum cele două hash-uri sunt comparate (cel obținut după decriptarea cu cheia publică cu cel obținut după aplicarea funcției de hash pe mesajul primit), iar dacă acestea sunt identice înseamnă că mesajul este într-adevăr trimis de emițătorul legitim și mai înseamnă că acest mesaj nu a fost modificat. [1]

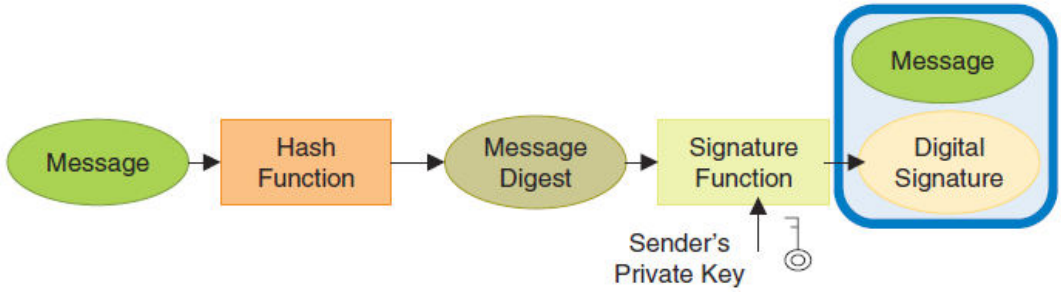


Figura 3.1 Crearea unei semnături digitale [1]

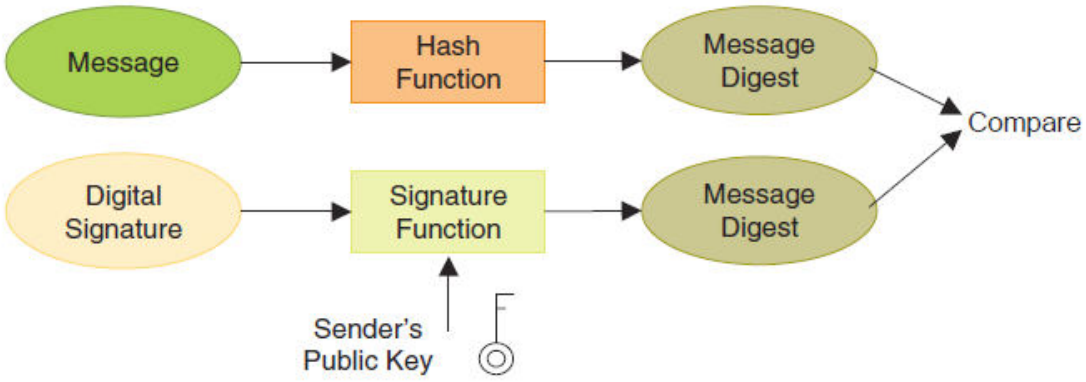


Figura 3.2 Verificarea unei semnături digitale [1]

Ca o concluzie, semnăturile digitale sunt reprezentate de structuri de date obținute în urma unor calcule criptografice efectuate asupra documentelor folosind cheia privată a celui care semnează. Integritatea și autenticitatea unui document semnat digital pot fi validate de oricine cu ajutorul cheii publice a semnatarului. Mai mult, se poate dovedi cine a semnat documentul. Unul sau mai mulți semnatari pot semna secvențial sau paralel un document electronic. Astfel, gestiunea documentelor electronice poate fi realizată asemănător documentelor în format fizic. Există niveluri diferite de semnătură digitală care pot fi atașate documentelor - semnături calificate pentru tranzacțiile electronice

care prin lege necesită o formă scrisă de semnătură și forme mai simple de semnătură pentru alte tranzacții care trebuie securizate. Încrederea în cheia publică a semnatarului este asigurată pe baza unui certificat digital emis de o Autoritate de Certificare (terț de încredere). [2]

3.4 Entități principale implicate într-o tranzacție bazată pe semnături electronice

Entitățile principale implicate într-o tranzacție bazată pe semnături electronice sunt [2]:

- ***Semnatarul*** - entitatea care creează semnătura electronica.
- ***Verificatorul*** - entitatea care verifică (validează) semnătura electronica.
- ***Furnizorul de servicii de încredere*** - furnizor de servicii electronice care asigură stabilirea unei relații de încredere între *Semnatar* și *Verificator.*
- ***Arbitrul*** - entitate competentă să arbitreze eventualele dispute care pot apărea între *Semnatar* și *Verificator.*

Regulamentul 910/2014 (Regulamentul eIDAS) [3] stabilește cadrul legal necesar furnizării serviciilor de semnătură electronică la nivelul Uniunii Europene [2].

Comisia Europeană de Normalizare (CEN) a identificat următoarele tipuri de semnături [2]:

1) ***semnături efemere*** - de interes pentru o perioadă foarte scurtă de timp, mai exact până la emiterea următoarei informații de revocare pentru certificatele implicate în procesul de validare.
2) ***semnături pe termen scurt*** - de interes pentru o perioadă de timp ce se încheie odată cu expirarea certificatului semnatarului sau a certificatelor din lanțul său de validare.
3) ***semnături pe termen lung*** - acestea trebuie să poată fi verificate și după expirarea certificatelor implicate în procesul de validare.
4) ***semnături pe termen foarte lung*** - ex. arhivare documente - sunt de interes și trebuie să poată fi verificate o perioadă de timp îndelungată după momentul semnării. De asemenea, aceste semnături trebuie să reziste deprecierii algoritmilor și cheilor criptografice folosite la momentul semnării.

3.5 Formate de semnătură digitală

3.5.1 Formatul CMS

Definit inițial de RSA Laboratories sub forma unei note tehnice de laborator cunoscută ca PKCS#7, CMS reprezintă formatul de semnătură care s-a impus în industrie [2]. Ultimul standard emis pentru acest format de semnătură este definit de RFC 5652 [4]. Conform [4], acest format este folosit pentru semnarea digitală, crearea unui hash, autentificarea sau criptarea unui mesaj cu conținut arbitrar. Standardele menționate definesc șase tipuri de mesaje criptografice [2]:

1) ***Data*** - descrie datele sub forma unui șir arbitrar de octeți interpretat de fiecare aplicație în parte.
2) ***Signed-Data*** - descrie modul în care poate fi asigurată integritatea și autenticitatea datelor prin folosirea semnăturilor digitale și a certificatelor digitale. Acest tip de mesaj reprezintă baza formatelor de semnătură digitală și este folosit și în soluțiile ETSI privind semnăturile electronice avansate. Figura 3.3 prezintă structura mesajului CMS de tip *Signed-Data*.
3) ***Enveloped-Data*** - descrie modul în care se poate asigura confidențialitatea datelor prin criptarea acestora cu o cheie de sesiune, cheie care este criptată pentru fiecare destinatar printr-un algoritm de distribuție de chei.
4) ***Digested-Data*** - descrie modul în care poate fi asigurată intergritatea datelor prin folosirea funcțiilor de hash.
5) ***Encrypted-Data*** - descrie modul prin care se poate asigura confidențialitatea mesajelor în urma criptării directe, fără folosirea cheilor de sesiune.
6) ***Authenticated-Data*** - descrie modul în care integritatea și autenticitatea datelor pot fi obținute prin utilizarea codurilor de autentificare a mesajelor (MAC).

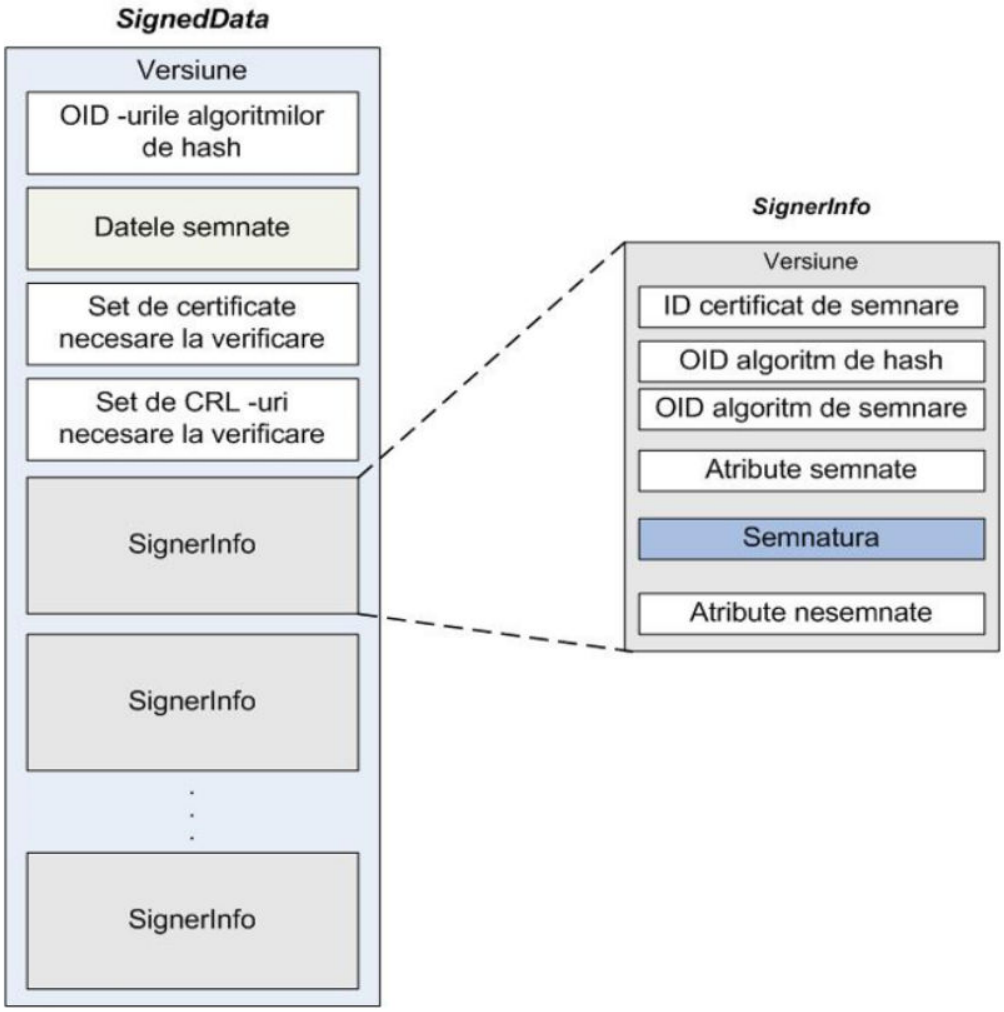


Figura 3.3 Structura mesajului CMS de tip *Signed-Data*

Formatul de semnătură CMS prezintă câteva probleme de securitate, și anume [2]:

1) **Este vulnerabil la atacul de substituție a certificatului semnatarului.**

   Mesajul CMS de tip *Signed-Data* nu face legătura în niciun fel între semnătura obținută asupra datelor și certificatul semnatarului (identificatorul certificatului semnatarului din structura *SignerInfo* nu este inclus în algoritmul criptografic de calcul al semnăturii). Astfel, dacă un utilizator are mai multe certificate emise pentru aceeași cheie publică, de exemplu un certificat este emis de o Autoritate de Certificare publică, iar altul de o Autoritate de Certificare organizațională, poate semna cu aceste certificate documente de valori diferite (poate semna cu certificatul emis de Autoritatea publică documente valabile în cadrul unei organizații ce necesită semnarea cu cel de-al doilea certificat). Mai mult, dacă pentru semnarea unui mesaj

este folosit un certificat expirat sau revocat (ceea ce duce la obținerea unei semnături invalide), iar apoi certificatul este înlocuit în *Signed-Data* cu unul valid, semnătura va fi și ea validată.

2) **Nu asigură protecția datelor de validare necesare la verificarea semnăturii.**

   Dacă o Autoritate de Certificare din lanțul de certificare este compromisă după momentul creării unei semnături, validitatea acestei semnături poate fi schimbată. Lanțul de certificare și informațiile de revocare pot fi modificate, un atacator putând substitui aceste informații în cadrul mesajului CMS fără ca acest lucru să poată fi detectat. Modificarea constrângerilor din cadrul căii de certificare existente la momentul semnării poate duce la schimbarea validității semnăturii.

3) **Nu asigură valabilitatea semnăturii pe termen lung și foarte lung.**

   Dacă la momentul semnării toate certificatele necesare erau valabile (neexpirate) și valide, semnătura trebuie să fie validată cu succes indiferent de starea certificatelor la momentul validării. Formatul CMS nu conține elemente de încredere care să permită evaluarea stării certificatelor în raport cu momentul semnării și nu cu momentul validării (standardul CMS prevede atributul *signing-time,* însă acest timp este generat de semnatar, ceea ce înseamnă că un semnatar poate să genereze o semnătură cu un timp anterior datei de revocare a certificatului său). Mai mult, formatul CMS nu oferă variante de răspuns pentru cazul în care cheile sau algoritmii de semnare devin slabi în timp și pot să nu mai corespundă normelor de securitate. Tot în contextul folosirii semnăturilor pe termen lung și foarte lung trebuie subliniat faptul că disponibilitatea și protecția datelor de validare (căi de certificare, informații de revocare) reprezintă aspecte foarte importante.

3.5.2 Formatele ETSI pentru semnătura electronică avansată

ETSI a elaborat trei seturi de standarde care reglementează formatul și îndeplinesc cerințele privind Semnătura Eletronică Avansată [2]:

- TS - 101 733 [5]: formatul CMS pentru Semnătura Electronică Avansată - CAdES.
- TS - 101 903 [6]: formatul XML pentru Semnătura Electronică Avansată - XAdES.

- TS - 102 778 [7]: formatul PDF pentru Semnătura Electronică Avansată - PAdES.

Standardele ETSI se bazează pe criptografia cu chei publice și pe certificatele digitale pentru generarea semnăturilor electronice. Valabilitatea pe termen lung a semnăturilor este asigurată prin folosirea tehnicilor de marcare temporală furnizate de terți de încredere (Autorități de Marcare Temporală) și arhivarea datelor critice (certificate, informații de revocare). Protejarea semnăturilor împotriva degradării algoritmilor criptografici și a cheilor pe termen lung este adresată prin marcarea temporală periodică. Specificațiile ETSI folosesc conceptul de politică de semnătură, această politică definind regulile ce trebuie respectate în procesul de creare și în cel de validare a unei semnături, astfel încât aceasta să poată fi acceptată ca fiind validă. [2]

3.5.2.1 Formatele CAdES

Scopul principal al acestui format este de a asigura valabilitatea pe termen lung și foarte lung pentru semnăturile electronice și de a rezolva problemele prezente la nivelul formatului CMS [2]. Practic, acest format are la bază CMS-ul plus câteva elemente necesare.

Specificația ETSI TS 101 733 [5] definește opt profiluri diferite pentru semnăturile electronice de tip CAdES:

1) **CAdES-BES** - asigură cerințele de bază pentru o semnătură electronică avansată. O semnătură de acest tip conține obligatoriu următoarele elemente: documentul semnat, o colecție de atribute semnate și semnătura. La nivelul atributelor semnate apar în mod obligatoriu următoarele 3 atribute: *Content-Type* - tipul datelor semnate, *Message-Digest* - rezumatul (hash-ul) datelor semnate și *Signing-Certificate* - acest atribut conține un rezumat calculat printr-o funcție de hash peste valoarea certificatului; se obține astfel protejarea împotriva atacurilor de substituție. Semnătura CAdES-BES asigură cerințele minime legale pentru semnăturile electronice avansate, dar nu asigură informații suficiente pentru ca semnătura să poată fi verificabilă pe termen lung [2].

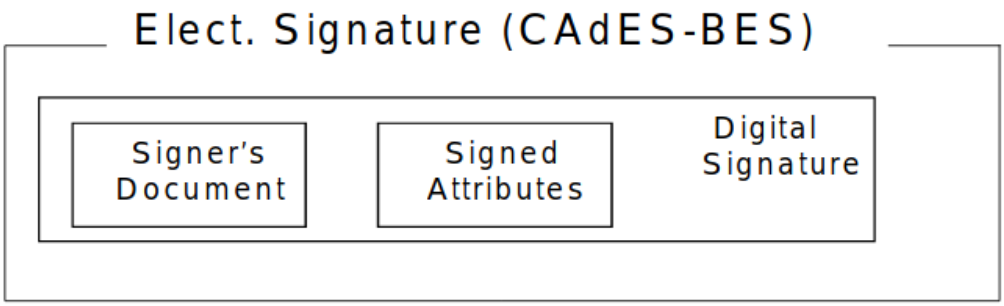


Figura 3.4 Profilul CAdES-BES [5]

2) **CAdES-EPES** - completează CAdES-BES și permite adăugarea unei politici de semnare [2].

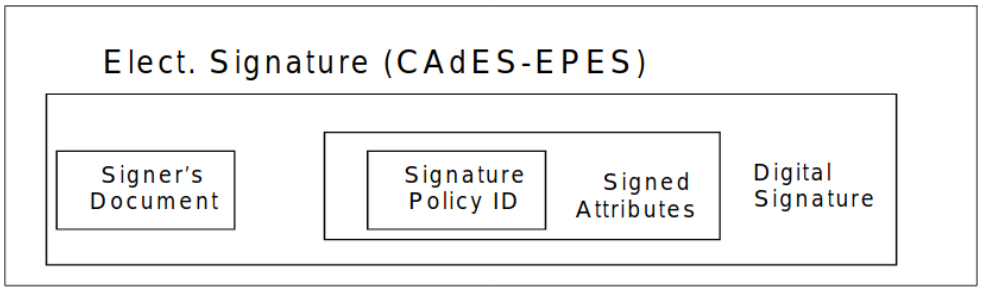


Figura 3.5 Profilul CAdES-EPES [5]

3) **CAdES-T** (Time-stamped) - completează CAdES-BES sau CAdES-EPES și permite adăugarea unei mărci temporale pentru a asigura nonrepudiabilitatea [2].

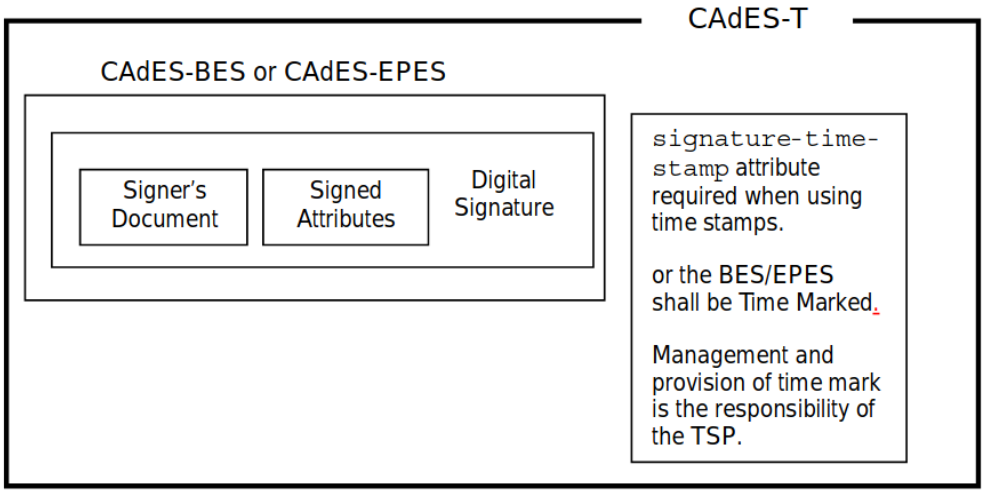


Figura 3.6 Profilul CAdES-T [5]

4) **CAdES-C** (Complete) - completează CAdES-T și permite adăugarea informațiilor de validare (referințe la certificatele din calea de certificare precum și la informațiile de revocare) asigurându-se astfel valabilitatea pe termen lung a semnăturii digitale [2].

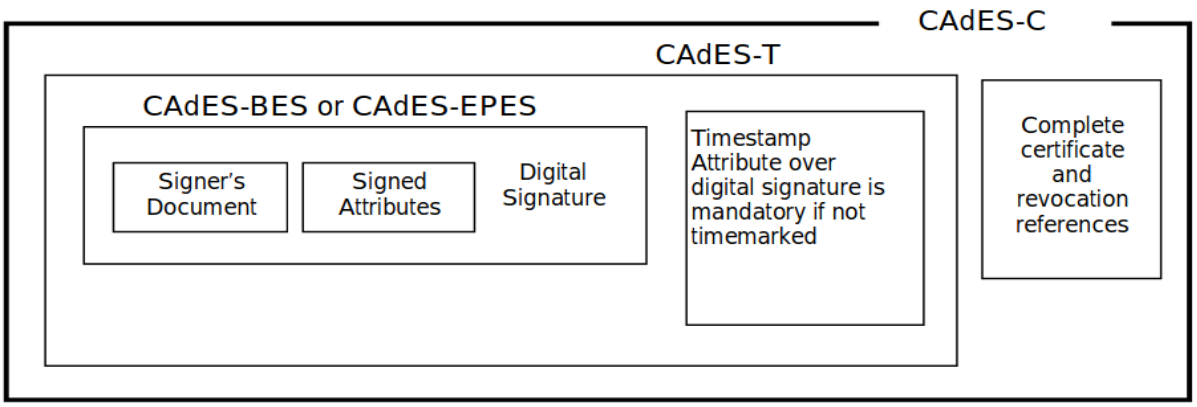


Figura 3.7 Profilul CAdES-C [5]

5) **CAdES-X** (Extended) - completează CAdES-C și permite protejarea informațiilor de validare prin marcarea temporală a referințelor acestora [2].

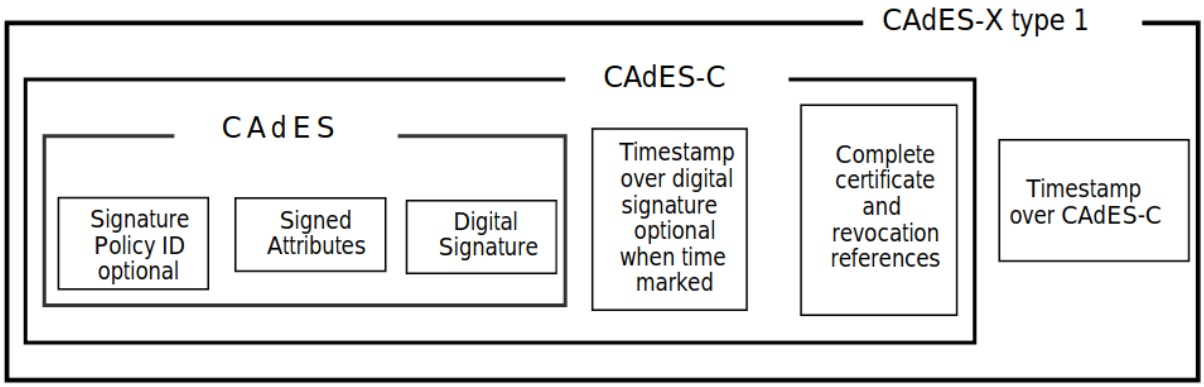


Figura 3.8 Profilul CAdES-X Tip 1 [5]

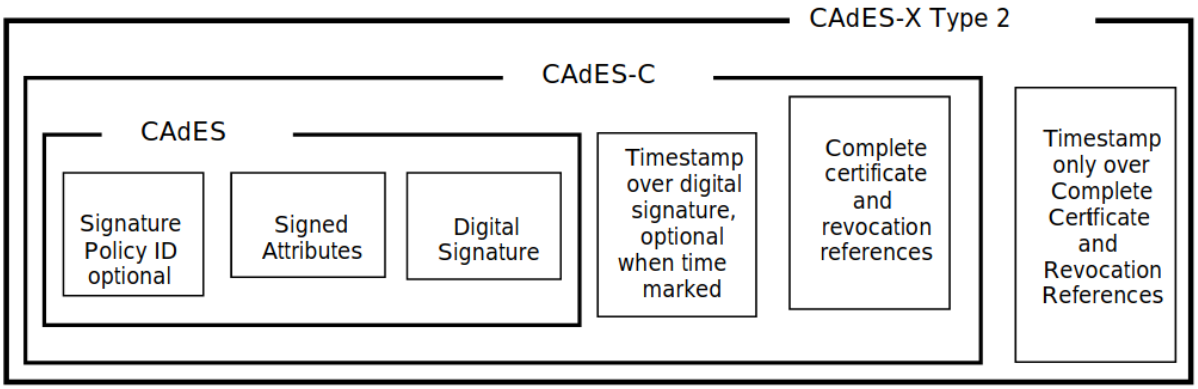


Figura 3.9 Profilul CAdES-X Tip 2 [5]

6) **CAdES-XL** (Extended Long) - completează CAdES-X și permite adăugarea informațiilor complete de validare (căile de certificare și informațiile de revocare necesare) [2].

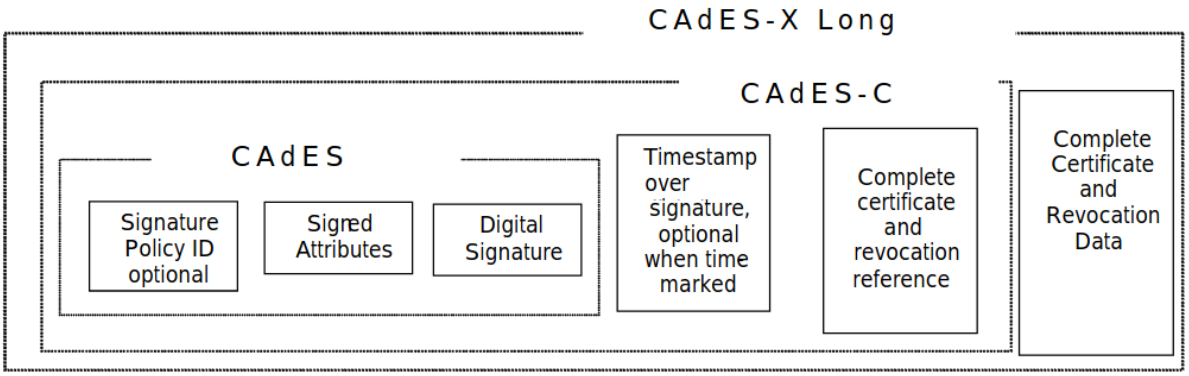


Figura 3.10 Profilul CAdES-X-Long [5]

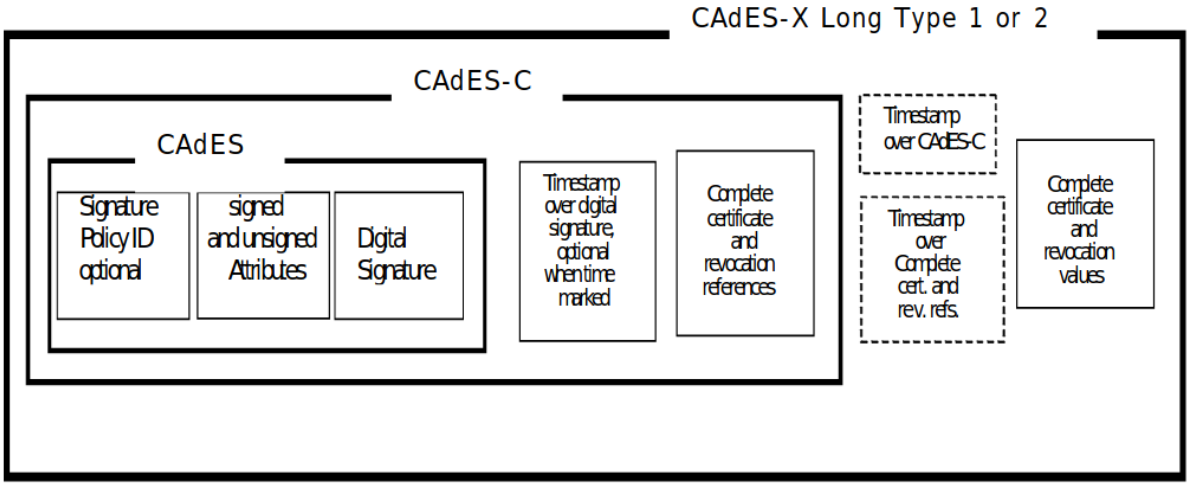


Figura 3.11 Profilul CAdES-X-Long Tip 1 și CAdES-X-Long Tip 2 [5]

7) **CAdES-A** (Archive) - completează CAdES-XL și permite protejarea periodică a informațiilor complete de validare prin (re)marcarea lor temporală, obținându-se astfel asigurarea valabilității semnăturii pe termen foarte lung [2].

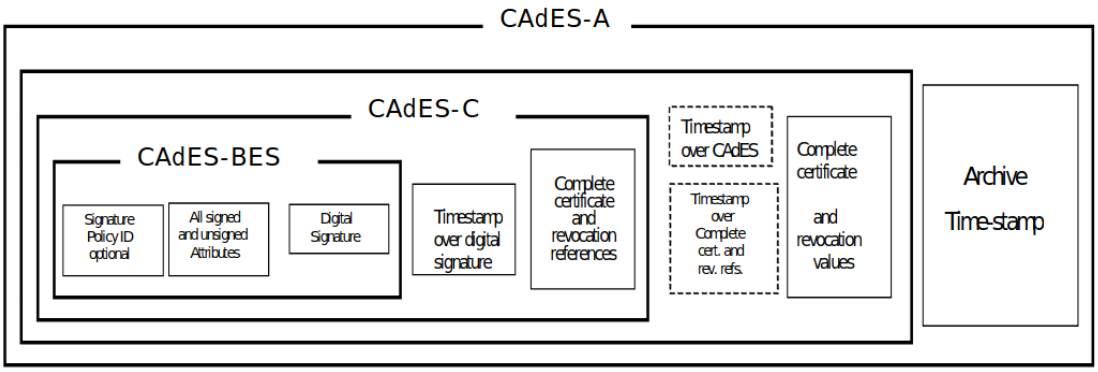


Figura 3.12 Profilul CAdES-A [5]

8) **CAdES-LT** (Long-Term) - se bazează pe orice format dintre CAdES-T, CAdES-C, CAdES-X Long, CAdES-Long Tip 1 sau 2 sau CAdES-A, permițând în plus adăugarea unuia sau mai multor atribute de validare pe termen lung (eng. *long-term-validation attributtes*).

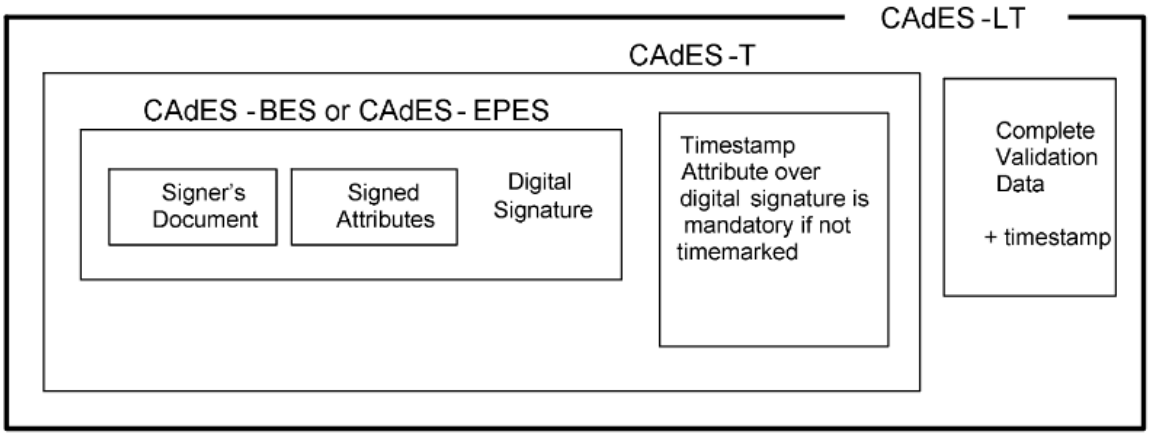


Figura 3.13 Profilul CAdES-LT [5]

3.5.2.2 Formatele XAdES

Aceste formate definite în TS 101 903 [6] constituie transpunerea formatelor de semnătură avansată CAdES în formatul semnăturilor XML [2].

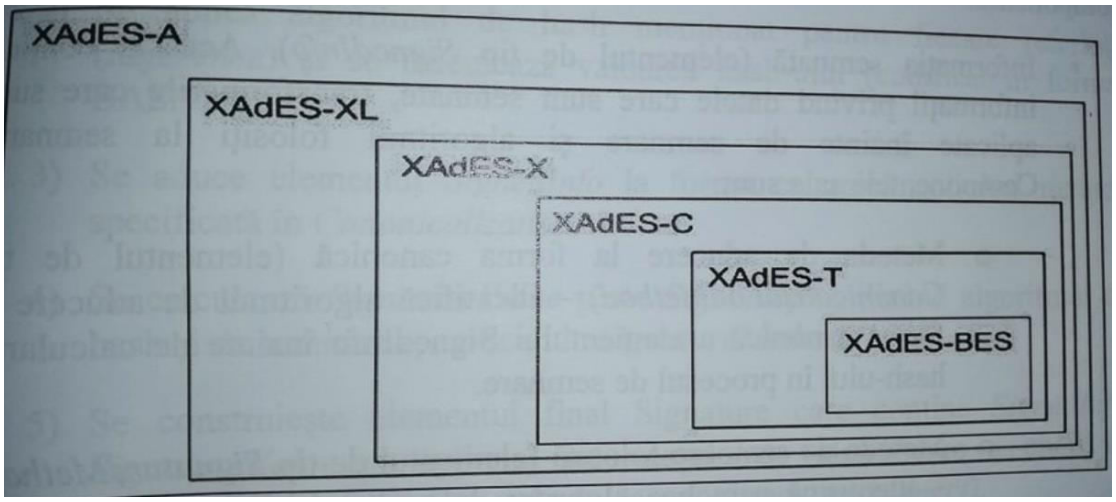


Figura 3.14 Nivelurile de semnătură XAdES [2]

Formatele XAdES extind formatul de semnătură XML propus de W3C în specificația [8]. Ideea de bază a semnăturilor XML și practic diferența față de semnăturile CMS este reprezentată de faptul că se pot semna doar anumite părți dintr-un document. Semnăturile XML pot fi aplicate peste orice tip de document adresabil prin URI.

O semnătură XML este în esență un element XML (în sensul definiției din specificația XML 1.0) ce are următoarea structură (? - zero sau o singură apariție, + - una sau mai multe apariții, * - zero sau mai multe apariții) [2]:

```
<Signature ID?>
  <SignedInfo>
    <CanonicalizationMethod />
    <SignatureMethod />
   (<Reference URI? >
     (<Transforms>)?
      <DigestMethod>
      <DigestValue>
    </Reference>)+
  </SignedInfo>
  <SignatureValue>
 (<KeyInfo>)?
 (<Object ID?>)*
</Signature>
```

Figura 3.15 Semnătură XML de bază [9]

Formatele XAdES extind semnătura XMLDsig (Figura 3.15) și implementează cerințele privind semnătura avansată. Cel mai simplu nivel de semnătură XAdES este XAdES-BES și asigură doar asocierea semnăturii cu semnatarul și cu cheia sa publică de validare a semnăturii. Idenitificarea cheii se face prin includerea în semnătură a certificatului folosind elementul de tip *SigningCertificate* (inclus ca proprietate semnată) sau folosind elementul de tip *KeyInfo* moștenit din XMLDsig. Formatul obținut, conduce desigur la evitarea atacului de substituție a certificatului semnatarului. Toate formatele XAdES sunt similare cu formatele CAdES și propun asigurarea acelorași obiective de securitate [2].

Figura 3.16 Structura XML pentru formatul de semnătură XAdES-BES [6].

3.5.2.3 Formatele PAdES

Formatele de semnătură PAdES reprezintă standardele ETSI privind semnăturile avansate aplicabile documentelor PDF. Standardul ISO 32000-1 conține specificațiile de reprezentare ale unui document PDF și modul prin care se adaugă semnături digitale la acest tip de documente. [2]

Familia de standarde ETSI TS 102 778 definește o serie de profiluri pentru semnăturile PDF. Semnăturile PAdES se bazează pe formatele CAdES și XAdES.

Referitor la semnăturile PAdES se pot sublinia următoarele aspecte [2]:

- permit semnarea sau validarea semnăturilor folosindu-se aplicații comune de gestiune a documentelor PDF (nu necesită aplicații de procesare speciale).
- furnizează o formă vizuală de reprezentare a semnăturii, aceasta putând constitui un avantaj de folosire pentru utilizatori.
- suportă semnături multiple și secvențiale.
- permit semnarea documentelor PDF, dar și a altui tip de conținut încapsulate în aceste documente (de exemplu, conținut XML).

Capitolul 4

Procesul de validare al semnăturilor digitale în conformitate cu standardele ETSI

4.1 Introducere

Semnăturile digitale reprezintă piatra de temelie pentru tranzacțiile electronice, cu condiția ca aceste semnături să fie declarate valide în urma unui proces de validare. Tranzacțiile electronice pot fi luate în considerare doar în contextul folosirii unor semnături valide, în orice alt caz putând fi considerate eșuate. Un participant la o tranzacție electronică poate face o cerere către un TSP pentru validarea unei semnături digitale. Acest TSP (Trust Service Provider) este numit SVSP (Signature Validation Service Provider) și oferă servicii de validare a semnăturii digitale. Rezultatul unui proces de validare este reprezentat de un raport de validare a semnăturii digitale. Participanții la o tranzacție electronică trebuie să aibă încredere că TSP-ul a implementat toate procedurile și măsurile de protecție în scopul minimizării riscurilor și ameniță rilor asociate semnăturilor digitale [10]. TS 119 441 [10] prezintă cerințele care trebuie îndeplinite de către furnizorii de servicii de încredere (TSP) în contextul validării semnăturilor folosite în tranzacțiile electronice și include, printre altele, cerințe ce se regăsesc în Regulamentul 910/2014 [3] (aceste cerințe stabilesc cadrul legal pentru crearea și validarea semnăturilor digitale).

4.2 Părți implicate în procesul de validare a semnăturii

Principalii actori prezenți în procesul de validare a unei semnături digitale sunt SVSP - furnizorul de servicii de validare a semnăturii (Signature Validation Service Provider care este un Trust Service Provider - furnizor de servicii de încredere) și clientul serviciului. Un SVSP poate oferi unul sau mai multe servicii de validare a semnăturii. Prin *utilizator* al serviciului de validare (practic, clientul serviciului) putem înțelege o aplicație ce realizează o cerere de validare sau o persoană ce folosește o aplicație pentru a crea o cerere de validare [10].

Alte părți implicate într-un proces de validare de semnătură digitală pot fi [10]:

- semnatarul - poate constrânge/limita semnătura printr-o politică de creare a semnăturii ceea ce poate influența procesul de validare.
- furnizorii de servicii de încredere **asociați semnatarului**:
  - CA-ul care a generat certificatul semnatarului
  - orice furnizor de servicii de încredere care poate fi implicat în procesul de generare a semnăturii:
    - TSP care gestionează un dispozitiv calificat de creare a semnăturii - (Q)SCD - în numele semnatarului
    - TSP care generează semnătura digitală.
    - autorități de marcare temporală (TSA).
    - etc.
- alți furnizori de servicii de încredere:
  - autorități de marcare temporală (TSA).
  - alți furnizori de servicii de validare a semnăturii (SVSP) către care furnizorul contractat inițial (SVSP) poate redirecționa o cerere.
  - etc.
- furnizori europeni și nu numai ce oferă liste de certificate ce pot fi considerate de încredere.
- comisia europeană ce furnizează LOTL [11]

4.3 Arhitectura unui serviciu de validare a semnăturii

Un serviciu de validare prevede următoarele componente:

- clientul serviciului de validare este o componentă software care implementează protocolul de validare a semnăturii la nivelul utilizatorului. În particular, acest client:
  - trimite o cerere de validare către server-ul serviciului de validare a semnăturii (SVSServ).
  - execută protocolul de validare a semnăturii (SVP) la nivelul utilizatorului.
  - în cazuri aplicabile, se ocupă de reprezentarea raportului de validare.
  - poate conține:

        - o interfață pentru încărcarea manuală a cererii.
        - un mod automat de trimitere a cererilor.
        - o interfață pentru reprezentarea raportului.

- server-ul serviciului de validare a semnăturii digitale (SVSServ) este o componentă care implementează protocolul de validare la nivelul SVSP. În particular, server-ul:
    - execută protocolul de validare a semnăturii și procesează validarea semnăturii la nivelul SVSP.
    - rulează aplicația de validare a semnăturii (SVA) (definită în ETSI TS 119 102-1 [12]) care implementează algoritmul de validare (definit în ETSI TS 119 102-1 [12]). În acest context, serviciul poate apela actori externi:
        - CA-ul care a semnat certificatul utilizatorului serviciului pentru obținerea informațiilor de revocare - OCSP, CRL.
        - CA-ul TSA-ului care a furnizat marca temporală din cadrul semnăturii.
        - un alt SVSP pentru verificări suplimentare.
        - etc.
    - creează raportul de validare a semnăturii asociat cererii.
    - trimite răspunsul de validare a semnăturii.

4.4. Pașii procesului de validare a semnăturii

1.Clientul generează și trimite o cerere de validare a semnăturii

Cererea include:

1) documentul semnat și semnătura aferentă; sau
2) o reprezentare a documentului semnat și semnătura, pentru a evita expunerea documentului către serviciul de validare.
3) (opțional) constrângerile de validare, după cum este definit în [12].

2.SVSServ efectuează procesul de validare

Procesul de validare este specificat în [12]. Validarea este realizată de către SVSP (Signature Validation Service Provider) conform constrângerilor care pot fi oferite de client sau de serviciul în sine:

1) dacă mulțimea de constrângeri nu este oferită de client, SVS (Signature Validation Service) poate implementa o politică implicită de validare.
2) dacă mulțimea de constrângeri este furnizată de client, atunci politica de validare a semnăturii oferită poate fi completată după cum cer practicile SVSP.

3.SVSServ pregătește și trimite răspunsul obținut în urma procesului de validare

Răspunsul validării conține raportul sau rapoartele de validare.

4.Reprezentarea raportului de validare

Aplicația client poate conține un modul de reprezentare a raportului de validare împreună cu alte informații relevante (vezi 5.2.9 din [12]). Bazându-se pe raportul de validare (ex. rezultat *indeterminate*), utilizatorul poate să accepte sau nu semnătura.

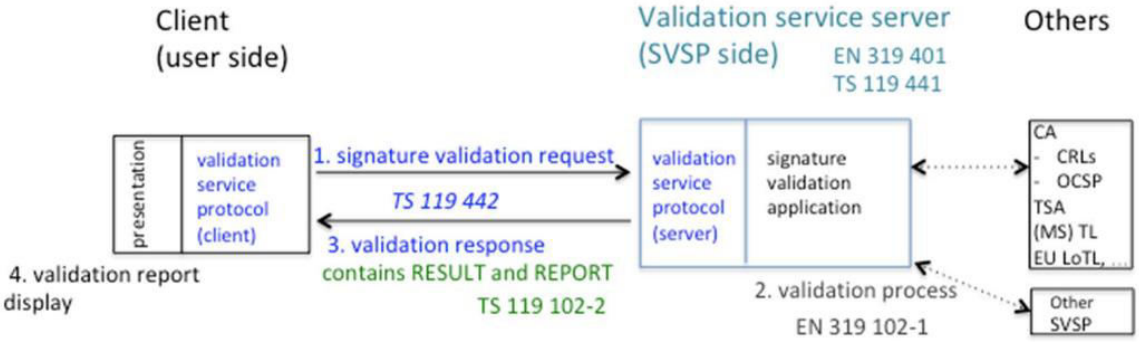


Figura 4.1 Pașii procesului de validare a semnături digitale [10]

4.5 Modelul conceptual de validare a semnăturii

Această secțiune definește modelul conceptual (Figura 4.2) de validare a semnăturii, împărțind un sistem de validare în două părți:

- SVA (eng. *Signature Validation Application*) - aplicația de validare a semnăturii; și
- DA (eng. *Driving Application*) - aplicaţia ce furnizează constrângerile sub care se desfăşoară procesul de validare şi semnătura în sine.

Aplicația de validare a semnăturii (SVA) primește la intrare o semnătură electronică avansată (AdES) și alte informații furnizate de DA.

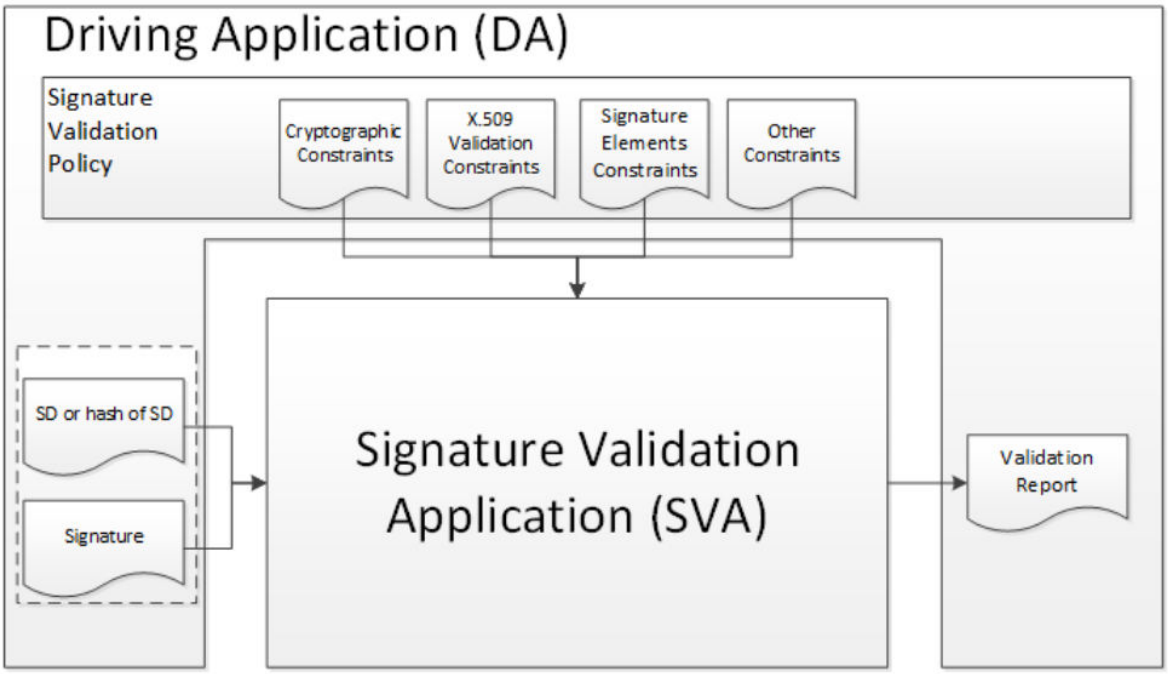


Figura 4.2 Modelul conceptual de validare a semnăturii [12]

SVA validează semnătura în raport cu o politică de validare a semnăturii, constând dintr-un set de constrângeri de validare și emite o indicație de stare și un raport de validare care furnizează detaliile tehnice de validare pentru fiecare constrângere aplicată, aceste informații fiind relevante pentru DA în acceptarea sau respingerea semnăturii.

Cu excepția cazului în care DA solicită SVA să execute un proces de validare specific, aceasta va începe întotdeauna cu procesul de validare pentru semnături ce furnizează disponibilitatea pe termen lung și integritatea datelor de validare (Def. 4 - SignatureLTAIVM, Anexa 1 Figura 5) (vezi 4.8.2.4). Primul pas al acestui proces este apelul validării pentru semnături cu marcă temporală (Def. 2, Anexa 1 Figura 3) și pentru semnături ce furnizează disponibilitatea pe termen lung a datelor de validare (Def. 3, Anexa 1 Figura 4) (vezi 4.8.2.3), proces care va apela la rândul lui validarea semnăturii de bază (Def. 1, Anexa 1 Figura 2) (vezi 4.8.2.1). În consecință, validarea urmează ciclul de viață al semnăturii (Anexa 1 Figura 1) și evaluează starea semnăturii plecând de la procesul de validare valabil pentru prima clasă de semnături (semnătura de bază) din ciclul de viață al semnăturii. Dacă procesul validării semnăturii de bază conduce la o concluzie definitivă (pozitivă sau negativă) întregul procesul de validare poate fi oprit. În alt caz, validarea continuă cu procesele specifice celorlalte clase de semnături (semnătură cu marcă temporală, SignatureLTVM, SignatureLTAIVM), până când poate fi trasă o concluzie definitivă sau nu mai există niciun proces de validare valabil. Rezultatul ultimului proces de validare aplicat este rezultatul final al validării (care poate fi de tip *rezultat nedeterminat* pentru cazul în care nu s-au oferit suficiente informații de validare) [12].

Pentru a se finaliza validarea unei clase de semnături, se aplică mai multe blocuri de validare (vezi 4.8.1). Indicația de stare rezultată în urma execuției fiecărui bloc de validare trebuie să aibă una dintre următoarele valori (fiecare clasă de semnături este validată în contextul unei politici de validare a semnăturii): PASSED, FAILED sau INDETERMINATE [12].

TOTAL-PASSED: rezultat obținut atunci când testele criptografice (inclusiv verificări ale hash-urilor obiectelor de date semnate indirect) au reușit și toate verificările descrise în politica de validare au reușit.

TOTAL-FAILED: rezultat obținut atunci când testele criptografice (inclusiv verificări ale hash-urilor obiectelor de date semnate indirect) au eșuat sau s-a demonstrat că certificatul folosit pentru semnare era invalid la momentul semnării sau pentru că semnătura nu a putut fi prelucrată de *blocul de verificare criptografică* (4.8.1.6) (semnătura nu este conformă cu unul dintre standardele de bază).

INDETERMINATE: rezultatul verificărilor efectuate nu permite să se constate că semnătura poate fi încadrată ca TOTAL-PASSED sau TOTAL-FAILED.

Rezultatul aplicației de validare a semnăturii este procesat în continuare de DA (pentru a fi prezentat celui care verifică semnătura). În acest context [12]:

- dacă SVA returnează TOTAL-PASSED pentru o anumită semnătura, DA ar trebui să considere semnătura ca fiind tehnic validă raportat la

constrângerile de validare (aceasta nu înseamnă că semnătura este utilă neapărat pentru un scop particular).

- dacă SVA returnează TOTAL-FAILED, DA nu ar trebui să considere ca fiind tehnic validă.
- dacă SVA returnează INDETERMINATE și există o indicație inferioară rezultatului principal care sugerează faptul că rezultatul se poate schimba prin reluarea algoritmului, DA ar trebui să ofere informații adiționale pentru validare sau să retrimită cererea validarea la un moment de timp ulterior. În orice alt caz, acceptarea semnăturii rămâne la latitudinea DA, respectiv a utilizatorului final. Tot în acest context, există cazuri când SVA nu poate procesa toate constrângerile de validare. De exemplu, dacă o politică de validare afirmă că timpul la care a fost realizată semnătura este timpul real chiar dacă nu există dovezi în acest sens, SVA nu poate lua în calcul în procesul decizional această considerație, returnând INDETERMINATE și indicând motivul. Aceasta va permite DA să accepte semnătura ca fiind validă conform politicii de validare oferite.

4.6 Indicațiile de stare ale procesului de validare a semnăturii

Aplicația de validare a semnăturii (SVA) trebuie să ofere un raport detaliat al procesului de validare, permițând DA să inspecteze modul în care s-au luat deciziile pe parcusul validării și să investigheze cauzele pentru indicația de stare rezultată.

Rezultatul procesului de validare trebuie să conțină [12]:

- indicația de stare rezultată în urma procesului (vezi Tabelul 4.1).
- o indicație care să specifice politica de validare utilizată sau constrângerile față de care s-a realizat procesul de validare.
- data și timpul la care a fost obținut rezultatul validării împreună cu informațiile de validare folosite.
- procesul de validare aplicat semnăturii.
- o indicație inferioară rezultatului principal (vezi Tabelul 4.2).
- date adiționale extrase din semnătură.

Indicațiile oferite de SVA trebuie să se conformeze următoarelor reguli:

- atunci când procesul de validare selectat returnează PASSED:
  - rezultatul per total trebuie să fie TOTAL-PASSED.
  - SVA trebuie să returneze raportul de validare asociat după cum este specificat în Tabelul 4.1.
- atunci când procesul de validare selectat returnează FAILED:
  - rezultatul per total trebuie să fie TOTAL-FAILED.
  - SVA trebuie să returneze o indicație inferioară rezultatului conformă cu Tabelul 4.2.
  - SVA trebuie să returneze rapoartele de validare asociate după cum este specificat în Tabelul 4.1 și în Tabelul 4.2.
- atunci când procesul de validare returnează INDETERMINATE:
  - rezultatul per total trebuie să fie INDETERMINATE.
  - SVA trebuie să returneze raportul de validare asociat după cum este specificat în Tabelul 4.1.
  - SVA trebuie să returneze motivul pentru un astfel de rezultat (motivul poate fi unul personalizat de către SVA sau unul din Tabelul 4.2.)

Tabelul 4.1 Indicațiile de stare ale procesului de validare [12]

Informația de validare raportată		**Semantică**
Indicația de stare	**Datele raportului de validare asociat**	
TOTAL-PASSED	Procesul de validare ar trebui să returneze lanțul de certificare validat, incluzând certificatul folosit la semnare. Mai mult, procesul de validare poate să ofere rezultatul validării	Rezultatul procesului de validare este TOTAL-PASSED pe baza următoarelor considerente: • verificarea de format a reușit; • verificările criptografice realizate asupra semnăturii au reușit; • orice constrângeri aplicabile

	pentru fiecare constrângere de validare în parte. Procesul de validare trebuie să ofere accesul DA la atributele semnate prezente în semnătura, identitatea semnatarului.	certificatului semnatarului au fost pozitiv validate; • semnătura a fost pozitiv validată în raport cu constrângerile de validare și prin urmare este considerată conformă cu aceste constrângeri.
TOTAL-FAILED	Procesul de validare ar trebui să ofere informații suplimentare în cazul indicației TOTAL-FAILED pentru fiecare dintre constrângerile de validare care au fost luate în considerare și pentru care s-a produs rezultatul negativ.	Rezultatul procesului de validare este TOTAL-FAILED pe baza următoarelor considerente: • verificările de format au eșuat; • verificările criptografice realizate asupra semnăturii au eșuat; • s-a dovedit că certificatul semnatarului nu era valid la momentul creării semnăturii.
INDETERMINATE	Procesul de validare ar trebui să ofere informații suplimentare în cazul indicației INDETERMINATE și să ajute verificatorul să identifice datele care lipsesc pentru finalizarea procesului de validare. În particular, trebuie să ofere indicații privind rezultatele validării pentru acele constrângeri care au fost luate în	Informațiile disponibile sunt insuficiente pentru a stabili dacă semnătura poate fi considerată TOTAL-PASSED sau TOTAL-FAILED.

	considerare și pentru care s-a produs un rezultat nedetermiant.	

Tabelul 4.2 Indicații secundare și semantica asociată [12]

Informația de validare raportată			**Semantică**
Indicația de bază	**Indicația secundară**	**Datele raportului de validare asociat**	
TOTAL-FAILED	FORMAT_FAILURE	Procesul de validare trebuie să ofere orice informație disponibilă pentru a se înțelege de ce analiza semnăturii a eșuat (formatul semnăturii).	Semnătura nu este conformă cu unul dintre standardele de bază, deoarece blocul de verificări criptografice nu poate procesa semnătura.
	HASH_FAILURE	Procesul de validare trebuie să ofere un identificator unic (URI sau OID) al elementului din cadrul obiectului de date semnat, care a cauzat eroarea.	Rezultatul procesului de validare este TOTAL-FAILED pentru că cel puțin un hash al unui obiect(e) de date semnat(e) nu corespunde valorii hash asociate la nivelul semnăturii.
	SIG_CRYPTO_FAILURE	Procesul de validare trebuie să returneze certificaul utilizat în procesul de	Rezultatul procesului de validare este TOTAL-FAILED deoarece valoare semnăturii nu a putut fi verificată în urma

		validare.	utilizării cheii publice din certificatul semnatarului.
	REVOKED	Procesul de validare trebuie să returneze următoarele: • lanțul de certificare folosit în procesul de validare. • timpul și dacă este accesibil, motivul revocării certificatului.	Rezultatul procesului de validare este TOTAL-FAILED deoarece: • certificatul folosit la semnare a fost revocat; și • poate fi dovedit faptul că semnătura a fost creată după momentul revocării.
	EXPIRED	Procesul de validare trebuie să returneze lanțul de certificate validat.	Rezultatul procesului de validare este TOTAL-FAILED deoarece poate fi dovedit că semnătura a fost creată după expirarea ceritifcatului folosit pentru semnare.
	NOT_YET_VALID	-	Rezultatul procesului de validare este TOTAL-FAILED deoarece poate fi dovedit că semnătura a fost creată înainte de emiterea certificatului folosit pentru semnare.
INDETERMINATE	SIG_CONSTRAINTS_FAILURE	Procesul de validare trebuie să returneze constrângerile care nu au fost îndeplinite de semnătură.	Rezultatul procesului de validare este INDETERMINATE deoarece unul sau mai multe atribute ale semnăturii nu se potrivesc constrângerilor de validare.

	CHAIN_CONSTRAINTS_FAILURE	Procesul de validare trebuie să returneze: • lanțul de certificate folosit în procesul de validare. • setul de constrângeri care nu au fost îndeplinite de lanțul de certificate.	Rezultatul procesului de validare este INDETERMINATE deoarece lanțul de certificate utilizat în procesul de validare nu se potrivește constrângerilor de validare referitoare la certificate.
	CERTIFICATE_CHAIN_GENERAL_FAILURE	Procesul de validare trebuie să furnizeze informații suplimentare în ceea ce privește motivul returnării unui astfel de rezultat.	Rezultatul procesului de validare este INDETERMINATE deoarece setul de certificate valabile pentru validarea lanțului de certificare a produs o eroare dintr-un motiv nespecificat.
	CRYPTO_CONSTRAINTS_FAILURE	Procesul de validare trebuie să returneze: • materialul (semnătură, certificat) care a fost produs folosindu-se un algoritm sau o dimensiune a cheii sub nivelul criptografic de securitate acceptat. • dacă este	Rezultatul procesului de validare este INDETERMINATE pentru că cel puțin unul dintre algoritmii folosiți pentru obținerea unei semnături, respectiv a unui certificat, sau dimensiunea unei chei folosite într-un astfel de algoritm sunt sub nivelul de securitate acceptat, și: • acest material (semnătură, certificat) a fost produs după timpul până la care acest

		cunoscut, timpul până la care algoritmul sau dimensiunea cheii erau considerate sigure.	algoritm/cheie a fost considerat sigur (dacă acest timp este cunoscut); și • materialul nu este protejat de o marcă temporală îndeajuns de puternică care să fi fost aplicată înainte de momentul până la care acest algoritm/cheie a fost considerat sigur.
	POLICY_PROCESSING_ERROR	Procesul de validare trebuie să furnizeze informații adiționale cu privire la rezultat.	Rezultatul procesului de validare este INDETERMINATE deoarece un fișier ce conținea politica de validare nu a putut fi procesat (nu a putut fi accesat, nu a putut fi parcurs, etc.)
	SIGNATURE_POLICY_NOT_AVAILABLE	-	Rezultatul procesului de validare este INDETERMINATE deoarece documentul electronic ce conține detaliile politicii de validare nu este disponibil.
	TIMESTAMP_ORDER_FAILURE	Procesul de validare trebuie să returneze lista de mărci temporale care nu respectă ordinea impusă la nivelul constrângerilor.	Rezultatul procesului de validare este INDETERMINATE deoarece constrângerile referitoare la ordinea mărcilor temporale sau/și a mărcilor temporale asociate obiectelor semnate nu sunt respectate.

	NO_SIGNING_CERTIFICATE_FOUND	-	Rezultatul procesului de validare este INDETERMINATE pentru că certificatul folosit pentru semnare nu poate fi identificat.
	NO_CERTIFICATE_CHAIN_FOUND	-	Rezultatul procesului de validare este INDETERMINATE deoarece nu a fost găsit niciun lanț de certificate pentru certificatul folosit la semnare.
	REVOKED_NO_POE	Procesul de validare trebuie să returneze următoarele informații: • lanțul de certificate folosit în procesul de validare. • momentul de timp și motivul revocării certificatului folosit pentru semnare.	Rezultatul procesului de validare este INDETERMINATE pentru că certificatul folosit era revocat la momentul validării. Cu toate acestea, algoritmul de validare a semnăturii nu poate stabili dacă momentul semnării se află înaintea sau după timpul de revocare al certificatului.
	REVOKED_CA_NO_POE	Procesul de validare trebuie să returneze următoarele informații: • lanțul de certificate ce include certificatul CA revocat.	Rezultatul procesului de validare este INDETERMINATE pentru că a fost găsit un lanț de certificate ce conține un certificat intermediar revocat.

		• momentul de timp și motivul revocării certificatului.	
	OUT_OF_BOUNDS_NOT_REVOKED	-	Rezultatul procesului de validare este INDETERMINATE pentru că certificatul semnatarului este expirat sau nu era încă valid la momentul validării, iar algoritmul de validare a semnăturii nu poate stabili dacă timpul semnării este cuprins în perioada de validitate a certificatului. Se știe că certificatul nu este revocat.
	OUT_OF_BOUNDS_NO_POE	-	Rezultatul procesului de validare este INDETERMINATE pentru că certificatul semnatarului este expirat sau nu era încă valid la momentul validării, iar algoritmul de validare a semnăturii nu poate stabili dacă timpul semnării este cuprins în perioada de validitate a certificatului.
	CRYPTO_CONSTRAINTS_FAILURE_NO_POE	Procesul de validare trebuie să returneze: • materialul (semnătură, certificat) care a fost produs folosindu-se un	Rezultatul procesului de validare este INDETERMINATE pentru că cel puțin unul dintre algoritmii folosiți pentru obținerea unei semnături, respectiv a unui certificat, sau dimensiunea unei chei

		algoritm sau o dimensiune a cheii sub nivelul criptografic de securitate acceptat. Dacă este cunoscut, timpul până la care algoritmul sau dimensiunea cheii erau considerate sigure.	folosite într-un astfel de algoritm, sunt sub nivelul criptografic de securitate acceptat și nu poate fi dovedit faptul că acest material (semnătură/certificate) a fost produs până în momentul la care acest algoritm/cheie a fost considerat sigur.
	NO_POE	Procesul de validare trebuie să furnizeze date adiționale.	Rezultatul procesului de validare este INDETERMINATE deoarece nu se poate determina o dovadă a existenței (POE) prin care să se stabilească dacă un obiect semnat a fost produs înainte de un eveniment compromițator (de exemplu, un algoritm criptografic a fost spart).
	TRY_LATER	Procesul de validare ar trebui să furnizeze un moment de timp la care este de așteptat ca datele de revocare să devină valabile.	Rezultatul procesului de validare este INDETERMINATE deoarece nu toate constrângerile pot fi verificate cu ajutorul informațiilor furnizate. Cu toate acestea, validarea poate fi posibilă prin folosirea unor date de revocare valabile la un moment de timp ulterior.

	SIGNED_DATA_NOT_FOUND	Procesul de validare trebuie să returneze un identificator al datelor semnate care a dus la apariția erorii, desigur dacă acest identificator este valabil (de exemplu, un URI).	Rezultatul procesului de validare este INDETERMINATE deoarece datele semnate nu pot fi obținute.

4.7 Constrângeri de validare

Un utilizator al aplicației de validare a semnăturii digitale trebuie să înțeleagă în detaliu cadrul în care se desfășoară un proces de validare a semnăturii digitale conform standardelor ETSI. În acest context, nu se face referire la procesul de validare în sine, ci mai degrabă la informațiile pe care un utilizator trebuie să le ofere aplicației de validare și desigur, la rapoartele de validare pe care aplicația le produce. Fără înțelegerea acestor aspecte, utilizatorul are mari șanse să greșească în acceptarea sau respingerea unei semnături. În secțiunea anterioară (4.6) au fost prezentate rezultatele (indicațiile de stare) pe care o aplicație de validare a semnăturii le furnizează în urma procesului de validare, iar în secțiunea curentă este subliniată importanța înțelegerii politicii de validare, în contextul în care un utilizator alege să furnizeze această politică (constrângerile de validare).

Procesul de validare este controlat de un set de constrângeri. Aceste constrângeri pot fi definite în următoarele moduri [12]:

- folosindu-se o politică definită la nivel formal după cum este specificat în [13];
- la nivelul sistemului în mod explicit: fișiere de proprietăți, constrângeri memorate în regiștri sau baze de date; sau
- în mod implicit de implementarea în sine.

De reținut este faptul că dacă un algoritm recomandă o anumită verificare, iar setul de constrângeri afirmă că acea verificare nu este necesară (de exemplu,

verificarea datelor de revocare pentru un certificat), o aplicație de validare a semnăturii trece peste acea verificare considerând-o reușită. În astfel de cazuri, SVA returnează, în raportul final, lista de verificări anulate de politica de validare [12].

Constrângerile suportate sunt următoarele:

- constrângeri de validare X.509 - aceste constrângeri trebuie să furnizeze cerințele pentru verificarea datelor de revocare și cerințele folosite în procesul de validare a căii certificatului conform cu [13], A.4.2.1, Tabelul A.2, linia m.
- constrângeri criptografice - aceste constrângeri trebuie să furnizeze cerințe cu privire la algortimii și parametrii folosiți la crearea semnăturilor sau folosiți la validarea obiectelor semnate conform cu [13], A.4.2.1, Tabelul A.2, linia p.
- constrângerile elementelor semnăturii - aceste constrângeri trebuie să furnizeze orice aspecte suplimentare constrângerilor anterior definite după cum este specificat în [13], A.4.2.1, Tabelul A.2.

4.8 Validarea propriu-zisă

4.8.1 Blocuri de bază

Acest subcapitol prezintă blocurile de bază care sunt ulterior folosite pentru construirea algoritmilor de validare executați în scenarii specifice. Figura 4.3 arată, într-o manieră simplificată, modul în care aceste blocuri sunt relaționate într-un proces de validare a semnăturii.

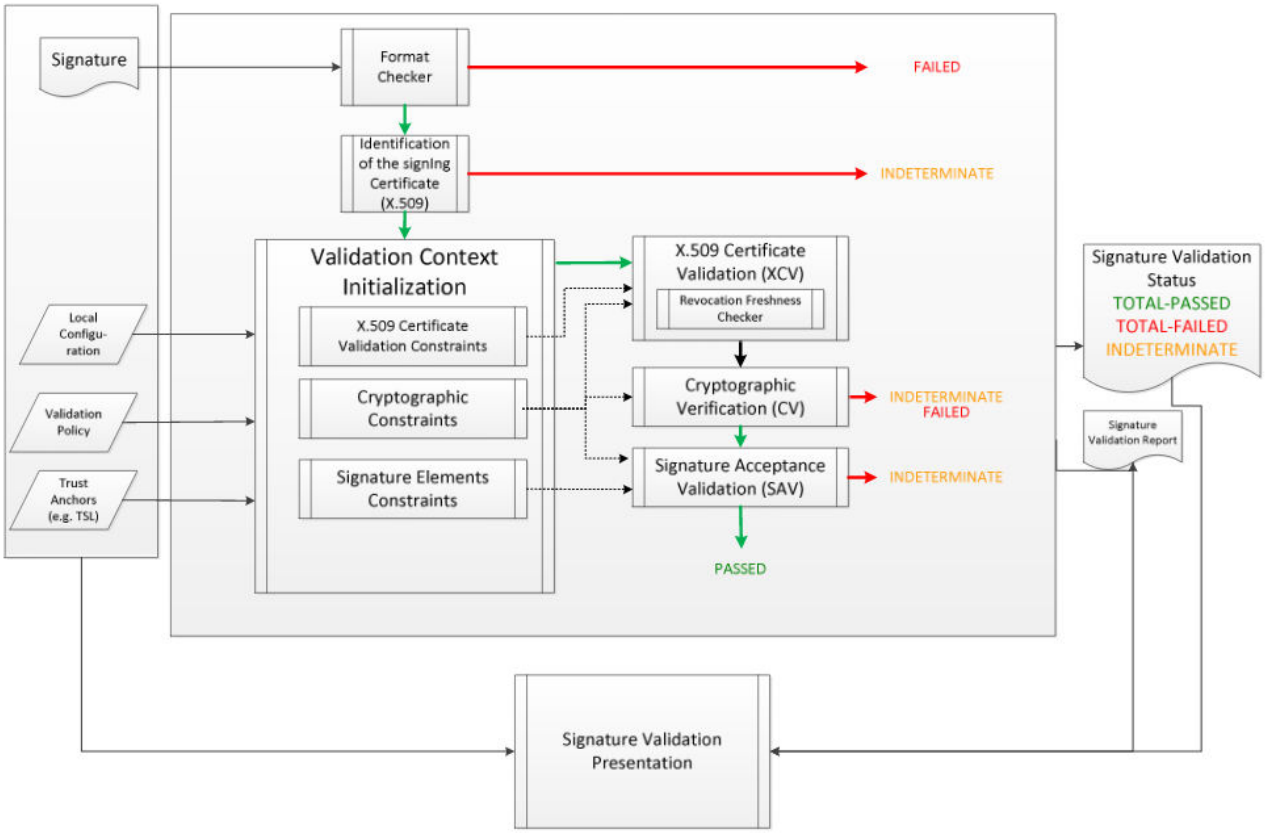


Figura 4.3 Validarea unei semnături de bază (*Basic Signature*) [12]

Acțiunile realizate de fiecare bloc în parte sunt următoarele [12]:

4.8.1.1 Verificarea formatului (*eng. Format Checking* realizată de blocul *Format Checker*) - acest bloc verifică dacă semnătura de validat este conformă cu formatul de bază aplicabil, în măsura în care conținutul său interior ar permite cel puțin să fie procesat de blocul de verificare criptografică. Această etapă nu include verificări care să ateste dacă o semnătură este de un anumit tip sau nivel, cum ar fi XAdES-E-XL sau PAdES-B-LTA. Dacă acest tip de verificări este cerut de politica de validare, poate fi inclus în *blocul de validare a acceptării semnăturii* (*SAV* - 4.8.1.7).

4.8.1.2 Identificarea certificatului folosit pentru semnare (blocul *Identification of the signing Certificate*) - acest bloc este responsabil cu identificarea certificatului ce va fi folosit la validarea semnăturii.

4.8.1.3 Inițializarea contextului de validare (blocul *Validation Context Initialization*) - acest bloc inițializează constrângerile de validare (constrângerile de validare X.509, constrângerile criptografice, constrângerile elementelor semnăturii) și parametrii corelați (parametri de validare X.509 incluzând lanțul de încredere și datele de revocare ale certificatului), aceste informații fiind folosite în procesul de validare a semnăturii.

4.8.1.4 Verificarea actualității datelor de revocare (blocul *Revocation Freshness Checker*) - acest bloc verifică dacă informația referitoare la starea revocării certificatului este „proaspătă” la un anumit moment de timp al procesului de validare. „Prospețimea” necesară a informațiilor privind starea de revocare este reprezentată de diferența maximă acceptată dintre momentul validării și momentul emiterii informațiilor de revocare. Acest proces este folosit și de alte blocuri de validare atunci când este necesară cunoașterea stării de revocare a certificatului. Acest bloc este important atunci când semnătura validată este o semnătură de bază (fără marcă temporală), iar momentul de timp furnizat este considerat insuficient. În astfel de cazuri, semnătura ar putea fi creată chiar înainte de începerea procesului de validare, dar cu toate acestea momentul exact de timp nu este cunoscut. Dacă datele referitoare la starea de revocare a certificatului sunt generate de foarte mult timp (nu sunt actuale) se poate ca certificatul să fi fost revocat înainte de crearea semnăturii, fapt ce nu poate fi observat din datele de revocare. În realitate, starea de revocare folosită este cea emisă cu puțin timp înaintea momentului curent, realizându-se astfel o aproximație a faptului că această informație este de încredere la momentul validării.

4.8.1.5 Validarea certificatului X.509 (blocul *XCV*) - acest bloc validează certificatul semnatarului la un moment de timp furnizat. Dacă momentul de timp pentru validare nu este furnizat, atunci validarea este realizată la momentul curent.

4.8.1.6 Verificarea criptografică (blocul *CV*) - acest bloc verifică integritatea datelor semnate prin aplicarea procedeelor criptografice.

4.8.1.7 Validarea acceptării semnăturii (blocul *SAV*) - acest bloc acoperă verificările suplimentare ce trebuie realizate pentru semnătura în sine sau pentru atribute existente la nivelul semnăturii.

4.8.1.8 Prezentarea validării semnăturii (blocul *Signature validation presentation*) - acest bloc este un element opțional și poate fi folosit pentru verificarea rezultatelor procesului de validare. Atunci când este prezent, blocul trebuie să furnizeze:

- prezentarea datelor (de exemplu, documentul semnatarului) care au fost acoperite de semnătură.
- prezentarea informațiilor ce identifică semnatarul.
- specificarea datei calendaristice și a momentului de timp în care a fost obținut rezultatul validării.
- prezentarea tuturor atributelor incluse în semnătură și realizarea clară a diferențierii între atributele semnate și atributele nesemnate.

- specificarea clară a politicii de validare folosită pe parcusul procesului.
- prezentarea rezultatului final al validării (TOTAL-PASSED, TOTAL-FAILED, INDETERMINATE).
- în cazul unui rezultat TOTAL-FAILED: prezentarea motivului ce a condus la invalidarea semnăturii.
- în cazul unui rezultat INDETERMINATE: sublinerea în raportul de validare a modificărilor necesare pentru obținerea unui rezultat determinat.
- prezentarea raportului de validare.

4.8.2 Procese de validare

4.8.2.1 Procesul de validare pentru semnături de bază

Acest proces are la bază blocurile de prelucrare descrise la 4.8.1 și este reprezentat în Figura 4.3. De asemenea, procesul în sine este folosit ca un bloc de bază pentru validarea mărcilor temporale (4.8.2.2.) și a semnăturilor cu marcă temporală (4.8.2.3). [12]

4.8.2.2 Procesul de validare al mărcilor temporale

O marcă temporală, după cum este definită în IETF RFC 3161 [14] și în ETSI EN 319 422 [15] [i.13], este o semnătură de bază. În acest context, procesul de validare al unei mărci temporale se construiește pe baza procesului descris la 4.8.2.1. [12]

4.8.2.3 Procesul de validare pentru semnături cu marcă temporală și pentru SignatureLTVM

SignatureLTVM diferă de semnăturile cu marcă temporală (*eng. Signature with Time*) printr-un set de informații suplimentare de validare ce sunt folosite în timpul procesului de validare. Astfel, procesul de validare este identic pentru aceste clase de semnături. Procesul începe (1) cu inițializarea mărcilor temporale extrase din atributele marcate temporal prezente în semnătură și cu inițializarea celui mai bun timp de semnare (*eng. best-signature-time* - variabilă

internă pentru algoritmul care indică primul moment de timp la care SVA poate să aibă încredere că o anumită semnătură a existat - acest moment este determinat printr-un POE existent la nivelul semnăturii sau este furnizat de DA și prin urmare este considerat de încredere). Procesul continuă (2) cu validarea semnăturii în sine (vezi 4.8.2.1), iar dacă semnătura conține materiale folosite pentru validarea pe termen lung, acestea sunt trimise separat de semnătură procesului descris la 4.8.2.1. Dacă (2) a reușit se trece la pasul (3) și anume, la validarea mărcilor temporale extrase (vezi 4.8.2.2), validarea realizându-se pentru fiecare marcă temporală în parte. După finalizarea analizei mărcilor temporale, SVA realizează (4) validarea acceptării semnăturii (vezi blocul *SAV* - 4.8.1.7) având ca date de intrare obiectul semnat, cel mai bun timp de semnare (vezi (1)) și constrângerile criptografice. Dacă (4) a reușit urmează (5): extragerea datelor ce conțin indicația de stare, lanțul de certificate și cel mai bun timp de semnare (vezi (1)).

4.8.2.4 Procesul de validare pentru SignatureLTAIVM

4.8.2.4.1 Blocuri de procesare adiționale

4.8.2.4.1.1 Blocul de validare la un moment de timp trecut al certifcatului

Acest proces validează un certificat la o dată/moment de timp care poate fi în trecut. Acest aspect poate deveni necesar la validarea pe termen lung în cazul în care un eveniment compromițător (de exemplu, expirarea certificatului CA-ului din lanțul de certificate) împiedică validarea obișnuită a certificatului (vezi blocul *XCV* – 4.8.1.5). În acest context, dacă un lanț de certificate a putut fi folosit pentru validarea unui certificat la un moment de timp trecut, atunci același lanț de certificate poate fi folosit în momentul curent pentru a se obține aceeași indicație de validitate, cu condiția ca fiecare certificat din lanț să îndeplinească unul din următoarele aspecte [12]:

1) starea de revocare a certificatului să poată fi stabilită la momentul curent (de obicei, dacă certificatul nu este încă expirat și se obțin informații privitoare la starea de revocare la momentul curent).
2) starea de revocare a certificatului poate fi stabilită prin folosirea informațiilor „vechi” de revocare dacă se poate dovedi că certificatul (respectiv, starea de revocare asociată) a existat la o dată în trecut, dată la care emitentul certificatului (respectiv, starea de revocare asociată acestuia) putea fi considerat de încredere și în controlul cheii sale private.

Astfel, acest proces va muta timpul de validare de la momentul curent la un anumit moment din trecut de fiecare dată când întâlnește un certificat care se dovedește a fi revocat, există o eroare produsă de constrângerile criptografice sau o eroare produsă de verificarea actualității datelor de revocare (vezi blocul *Revocation Freshness Checker* – 4.8.1.4). (vezi *procesul de șiftare a timpului de validare* – 4.8.2.4.1.2). Pe lângă returnarea lanțului de certificate, acest proces va returna ultima valoare a timpului de validare asociată certificatului țintă, valoare ce reprezintă un moment de timp la care certificatul este valid și lanțul de certificate poate fi validat. Orice obiect pentru care se demonstrează că a fost semnat cu certificatul țintă înainte de ultima valoare a timpului de validare poate fi acceptat ca un obiect valid. Această ultimă afirmație stă la baza proceselor de verificare prezentate în secțiunile următoare. [12]

4.8.2.4.1.2 Procesul de șiftare a timpului de validare

Acest bloc de procesare șiftează timpul de validare de la momentul curent la un anumit moment din trecut de fiecare dată când întâlnește un certificat care se dovedește a fi revocat, există o eroare produsă de constrângerile criptografice sau o eroare produsă de verificarea actualității datelor de revocare (vezi blocul *Revocation Freshness Checker* – 4.8.1.4). Procesul returnează ultima valoare a timpului de validare asociată certificatului țintă, valoare ce reprezintă un moment de timp la care certificatul este valid și lanțul de certificate poate fi validat. (1) Procesul va inițializa *timpul de control* (variabilă internă folosită în cadrul algoritmului ce nu face parte din rezultatul procesului de validare) ca fiind timpul/data curentă. (2) Pentru fiecare certificat din lanțul de certificare se vor extrage informațiile de revocare (aceste informații trebuie să fi fost emise înainte de *timpul de control*). Acum fiecare informație de revocare este analizată separat - dacă certificatul este marcat ca fiind revocat *timpul de control* va deveni timpul de revocare, iar dacă nu este marcat ca fiind revocat se va verifica actualitatea datelor de revocare (blocul *Revocation Freshness Checker* – 4.8.1.4) - dacă acest proces returnează FAILED se va seta *timpul de control* ca fiind timpul emiterii datelor de revocare, în caz contrar acest *timp* rămânând neschimbat (timpul curent). În continuare se aplică politicile criptografice specifice unui certificat - dacă certificatul îndeplinește aceste constrângeri *timpul de control* va deveni ultimul moment de timp până la care toți algoritmii erau considerați de încredere. De subliniat este faptul că (2) aplică acești pași pentru fiecare certificat în parte, respectiv pentru fiecare informație de validare în parte. La final procesul trebuie să întoarcă PASSED și *timpul de control* calculat dacă nu au existat evenimente care să oprească procesul pe parcurs (de exemplu, nu s-a găsit niciun POE pentru un certificat).

4.8.2.4.1.3 Extragerea POE

Acest proces derivează POE dintr-o marcă temporală primită la intrare. Se presupune că procesul de validare a mărcii temporale a reușit. Astfel, în cazul cel mai simplu, crearea unei mărci temporale oferă un POE pentru fiecare obiect marcat în momentul creării acestei mărci. De exemplu, o marcă temporală aplicată unei semnături demonstrează existența semnăturii la momentul creării mărcii de timp [12]. Cu alte cuvinte, chiar dacă semnătura a existat de dinainte, ea este considerată ca existând din acest moment - momentul creării mărcii temporale aferente. Astfel se obțin dovezile de existență (POE) ale obiectelor semnate.

4.8.2.4.1.4 Blocul de validare în trecut a semnăturii

Acest bloc este folosit atunci când validarea unei semnături (sau a unei mărci temporale) eșuează la momentul curent cu un rezultat de tip INDETERMINATE. POE furnizate acestui bloc pot ajuta la stabiliarea unui rezultat determinat. Astfel, acest bloc aplică procesul de validare în trecut a certificatului (4.8.2.4.1.1) cu următoarele intrări: semnătura, certificatul țintă, parametri de validare X.509, datele de validare, constrângerile X.509, constrângerile criptografice și setul de POE. Dacă acest proces nu a reușit blocul curent de validare a semnăturii întoarce timpul curent și motivul ce explică eroarea apărută. Dacă procesul de validare în trecut a certificatului întoarce PASSED/timpul de validare, POE pentru semnătură este raportat la timpul de validare primit. Astfel, în funcție de anumite indicații secundare rezultatului inițial INDETERMINATE, indicații precum REVOKED_NO_POE, REVOKED_CA_NO_POE, OUT_OF_BOUNDS_NO_POE, etc. procesul poate întoarce PASSED pentru semnătura verificată. În caz de eroare procesul întoarce motivul erorii. [12]

4.8.2.4.2 Procesul de validare propriu-zis pentru SignatureLTAIVM

Acest proces are la bază procesele de validare pentru semnături cu marcă temporală și pentru SignatureLTV (4.8.2.3), folosindu-se în plus de blocurile adiționale prezentate anterior. Acest proces are nevoie în mod obligatoriu de obiectul de date semnate, dar poate primi și informații opționale precum datele

de validare, certificatul folosit la semnare, etc. La final întoarce un rezultat ce indică validitatea semnăturii. [12]

Capitolul 5

Sistemul creat pentru validarea semnăturii digitale conform standardelor ETSI

5.1 Arhitectura sistemului

5.1.1 Tehnologii utilizate și părți componente ale sistemului

În continuare sunt enumerate și descrise tehnologiile folosite și elementele ce compun sistemul:

5.1.1.1 DSS (Digital Signature Service) [16]

Librărie open-source folosită pentru crearea şi validarea semnăturilor electronice în conformitate cu legislaţia europeană. În particular, DSS urmează regulamentele şi standardele eIDAS. Documentul [16] prezintă câteva exemple care demonstrează cum se poate implementa o aplicație în limbajul Java folosindu-se framework-ul DSS. Scopul este de a arăta dezvoltatorilor, în mod progresiv, diferitele utilizări ale framework-ului, astfel încât aceștia să se familiarizeze cu metodele de implementare puse la dispoziție. Exemplele prezentate urmează versiunea 5.5 a framework-ului care se găsește la [17]. La nivelul framework-ului pot fi diferențiate următoarele funcționalități de bază: crearea unei semnături digitale, extinderea unei semnături digitale și validarea unei semnături digitale. În document mai sunt detaliate formatele documentelor semnate (XML, PDF, DOC, TXT, ZIP), modurile de împachetare ale semnăturii (semnătură înfășurată, detașată, etc.), formatele semnăturilor digitale (XAdES, PAdES, CAdES și ASiC-S/ASiC-E), modul în care se prelucrează datele de revocare, modul în care se construiește lanțul de certificate, etc. [16]

5.1.1.2 Spring Boot

Framework open-source bazat pe Java și folosit pentru dezvoltarea de microservicii. Este dezvoltat de echipa Pivotal şi este folosit pentru dezvoltarea diferitelor tipuri de aplicații: aplicaţii *stand-alone* (pot fi limitate la rularea pe un singur calculator), aplicaţii folosite pe scară largă (sau în medii de producție complexe) [18].

Un microserviciu este o componentă a unei arhitecturi software ce permite dezvoltatorilor să creeze şi să ruleze servicii în mod independent. Fiecare serviciu rulează într-un proces separat construindu-se astfel aplicaţii utilizate pe scară largă şi care sunt totuşi uşor de administrat. [18]

Acest framework a fost ales deoarece pune la dispoziţie componente pentru crearea de servicii REST (REST *endpoints*/REST API) şi este compatibil

cu framework-ul DSS (este implementat tot în Java, iar dependinţele sunt de tip maven şi sunt adăugate într-un fişier pom.xml).

5.1.1.3 NginX

Server web ce poate fi folosit și ca *reverse proxy*, *load balancer*, *mail proxy* şi *HTTP cache* [19]. Această componentă este folosită ca *reverse proxy*, primind cererile de validare (apelurile endpoint-urilor) peste *https* (nginx-ul este folosit ca punct de intrare în reţeaua internă - vezi Figura 5.4) şi redirecţionându-le pe *localhost* peste *http*. Această abordare a fost aleasă deoarece seriviciul de validare dezvoltat rulează într-un server Tomcat încorporat (*eng. embedded*) la nivelul aplicației Spring Boot.

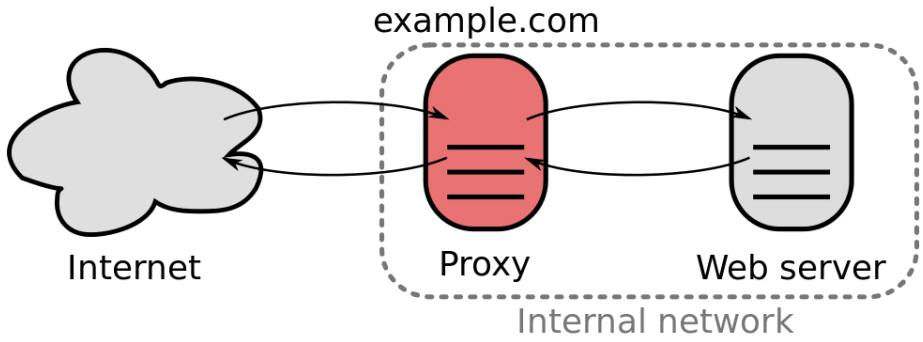


Figura 5.1 Schemă de reverse proxy [20]

5.1.1.4 Postman [21]

Program folosit pentru apelul şi testarea endpoint-urilor, practic componenta ce va realiza cererea de validare conform Figurii 5.4 (clientul serviciului).

5.1.1.5 Server ce furnizează informațiile privind starea de revocare a certificatelor

Poate fi un server web sau un server de aplicații. În cazul de față s-a folosit un server de aplicații **Tomcat**. Acesta a fost configurat ca **serviciu de**

linux, începându-și execuția odată cu pornirea sistemului de operare. Trebuie subliniat că se poate folosi orice server atât timp cât oferă acces la lista de certificate revocate. Practic, pe acest server s-a construit CA-ul și tot ce ține de crearea, modificarea, revocarea certificatelor folosite pentru semnare, testare etc. Atunci când certificatele sunt generate (s-a creat un script de generare automată a certificatelor utilizatorilor cu scopul de a ușura crearea certificatelor folosite pentru testarea sistemului; s-a implementat un script pentru generarea certificatului server-ului de OCSP), în fișierul de configurare (*rootca.conf*) al autorității de certificare sunt folosite extensii pentru certificate (practic ce acțiuni pot să se întreprindă prin folosirea respectivului certificat și a cheii private asociate) precum: *digitalSignature, nonRepudiation, keyEncipherment* (cunoscute ca *keyUsage*), *clientAuth*, *serverAuth* (cunoscute ca *extendedKeyUsage*). La nivelul acestor extensii sunt setate și sursele pentru informațiile de revocare: *authorityInfoAccess* - ce conține *aia_url (*URL-ul către certificatul CA-ului; AIA - Authority Information Access*)* și *ocsp_url (*URL-ul către domeniul server-ului de OCSP*)*; și *crlDistributionPoints* ce conține calea către locația de pe server unde se găsește lista de certificate revocate.

Acest server a fost configurat la adresa http://localhost:8080/.

CA-ul a fost implementat și configurat folosindu-se *openssl [22]*.

51.1.6 Server de OCSP

Server de OCSP care furnizează starea de revocare a unui certificat la momentul curent (pentru detalii privitoare la OCSP vezi [23]). Server-ul a fost configurat la adresa http://localhost:8888 folosindu-se *openssl [22]*. Pentru pornirea server-ului este folosit certificatul generat de CA pentru OCSP și certificatul CA-ului.

5.1.1.7 OpenSSL

OpenSSL este reprezentat de un set de instrumente open-source folosite în interacțiunea cu protocoalele TLS și SSL. De asemenea, este o librărie criptografică folosită pentru dezvoltarea de aplicații ce utilizează algoritmi criptografici. [22]

5.1.1.8 HikariCP [24] și HSQLDB [25]

HikariCP este un *pool* de conexiuni JDBC [24]. HSQLDB este o bază de date relațională care se conformează standardului SQL: 2011 și specificațiilor

JDBC 4 [26]. Aceste tehnologii au fost utilizate pentru crearea unei baze de date în memoria principală (RAM) (cunoscută ca *in-memory database - IMDB*) și pentru conexiunea la această bază de date. În Figura 5.2 se pot vedea parametri de configurare pentru HSQLDB, iar în Figura 5.3 se poate vedea implementarea conexiunii de tip HikariCP. Acest sistem de persistență a fost implementat pentru crearea unui *cache* ce reține lista de certificate revocate (CRL) o anumită perioadă de timp definită de administratorul SVA. După acest timp, CRL-ul este din nou descărcat și reținut în *cache*.

```
#datasource config paramters
datasource.driver.class = org.hsqldb.jdbcDriver
datasource.url = jdbc:hsqldb:mem:svamemdb
datasource.username = sa
datasource.password =
```

Figura 5.2 Parametri de configurare HSQLDB

```
@Bean
public DataSource dataSource() {
    HikariDataSource ds = new HikariDataSource();
    ds.setPoolName("SVA-Hikari-Pool");
    ds.setJdbcUrl(dataSourceUrl);
    ds.setDriverClassName(dataSourceDriverClassName);
    ds.setUsername(username);
    ds.setPassword(password);
    ds.setAutoCommit(false);
    return ds;
}
```

Figura 5.3 Implementare “pool” de conexiuni de tip HikariCP

5.1.2 Interacțiunea componentelor sistemului. Pașii parcuși pentru validarea semnăturii

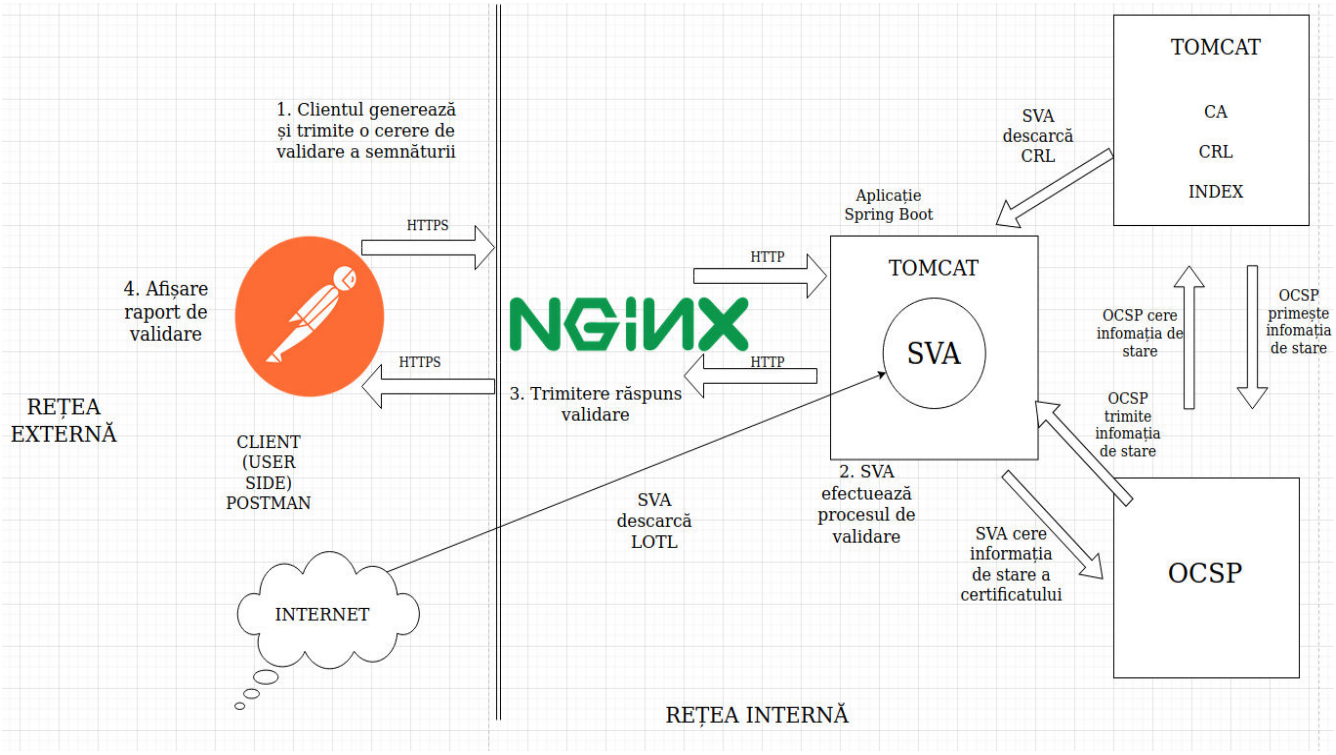


Figura 5.4 Interacțiunea componentelor. Procesul de validare

Administratorul serviciului de validare are acces la fișierul de proprietăți al aplicației (application.properties); acest fișier permite modificarea parametrilor aplicației.

La pornirea serviciului de validare a semnăturii sunt parcurși următorii pași:

1) se stabilește portul pe care va rula serviciul - proprietatea **server.port** din fișierul de proprietăți, astfel portul poate fi modificat înainte de pornirea aplicației.

2) se încarcă lista de certificate considerate de încredere. Aceste certificate sunt păstrate într-un fișier **keystore.p12**. Un administrator al serviciului poate furniza fișierul cu certificate considerate de încredere și parola aferentă fișierului. În cazul de față, fișierul de tip keystore conține certificatul autorității de certificare create (root-ca.crt) și certificatele ce se găsesc în jurnalul oficial al Uniunii Europene [27].

   La data de 16 octombrie 2009, Comisia Europeană a adoptat o decizie ce stabilește măsuri care facilitează utilizarea procedurilor bazate pe mijloace electronice prin „punctele unice de contact” (Def. 10). Una dintre măsurile adoptate prin decizie a constat în obligația statelor membre de a stabili și publica până la 28.12.2009 lista de încredere a furnizorilor de servicii de certificare acreditați care eliberează certificate

calificate pentru public. Obiectivul acestei măsuri este de a spori utilizarea transfrontalieră a semnăturilor electronice prin creșterea încrederii în semnăturile electronice provenite din alte state membre. Decizia a fost actualizată de mai multe ori din 16.10.2009; ultimul amendament a fost făcut la 01.02.2014. Cu scopul de a permite accesul ușor la lista de certificate considerate de încredere (practic lista furnizorilor de certificate ai statelor membre EU), Comisia Europeană a publicat o listă centralizată. Această listă este cunoscută sub abrevierea LOTL. [16] Lista este publicată la [28]. Această listă trebuie să fie semnată de un certificat considerat de încredere, iar pentru a ști cine are permisiunea de a semna/publica LOTL se vizualizează jurnalul oficial al EU [27]. Dacă semnătura acestei liste este validă atunci conținutul poate fi considerat de încredere [16]. În cazul aplicației implementate, se verifică semnătura LOTL, iar dacă rezultatul este pozitiv LOTL este considerată de încredere.

3) se descarcă LOTL și se adaugă la lista certificatelor considerate de încredere (pas ilustrat și în Figura 5.4).

Procesul de validare are loc în mai multe etape descrise la 4.4. Aceste etape au fost luate în calcul și la momentul implementării serviciului de validare. În acest sens, procesul de validare din Figura 5.4 poate fi descris astfel:

1) clientul serviciului (aplicația Postman/utilizatorul) generează și trimite o cerere de validare a semnăturii.

2) cererea este trimisă folosindu-se protocolul https. Practic, server-ul de NginX, care în cazul nostru joacă rolul de *reverse proxy*, se autentifică pe bază de certificat la aplicația client. Acest certificat este generat de CA-ul configurat la nivelul server-ului de Tomcat. NginX trimite cererea mai departe către serviciul de validare a semnăturii.

3) serviciul de validare a semnăturii (SVA) primește cererea de validare peste protocolul http. În acest moment începe procesul de validare a semnăturii după cum este descris la 4.8. Pe parcusul procesului, după cum a fost explicat la partea teoretică, sunt verificate datele de revocare aferente certificatului folosit pentru semnare. În acest sens, serviciul implementat extrage de la nivelul ceritificatului furnizat (practic se uită la nivelul extensiilor certificatului) sursele unde se găsesc datele de revocare (ocsp_url, aia_url - pentru construirea lanțului de certificate, crl_url). În funcție de datele obținute serviciul se comportă după cum urmează:

a) dacă nu a fost găsit niciun URL pentru OCSP se trece la pasul (b) în caz contrar (dacă URL-ul a fost găsit) se realizează o cerere către serverul de OCSP pentru aflarea stării de revocare a certificatului. Dacă server-ul de OCSP nu este disponibil (nu poate fi accesat) se trece la pasul (b), în caz contrar, rezultă că server-ul a trimis un răspuns (informație privind starea de revocare) care este analizat. Analiza este realizată conform proceselor de validare prezentate la partea teoretică și a tipului de semnătură furnizată (semnătură de bază, semătură cu marcă temporală, SignatureLTV, SignatureLTAIVM). Simplificând, în funcție de rezultatul final: certificat valid, revocat sau stare necunoscută, se continuă sau nu procesul de validare. Dacă procesul este oprit se întoarce un rezultat descris în Tabelul 4.2. Concluzionând, dacă server-ul de OCSP a putut fi contactat și a întors un rezultat nu se mai verifică lista de certificate revocate.

b) dacă nu este găsit un URL pentru CRL se întoarce un rezultat descris în Tabelul 4.2 și procesul se încheie (desigur după terminarea verificărilor în funcție de procesul de validare selectat). Dacă este găsit un URL pentru CRL se descarcă lista de certificate revocate și se adaugă în *cache* (5.1.1.8). Acum se caută informațiile de revocare pentru certificatul folosit la semnare și în funcție de informațiile obținute (starea de revocare) procesul continuă sau se încheie.

4) SVA generează și trimite rezultatul procesului de validare.

5) se afișează raportul de validare pe client. Este de subliniat faptul că aplicația returnează patru tipuri de rapoarte de validare, fiecare conținând rezultatele validării cu mai multe sau mai puține detalii. Astfel, se returnează raportul simplu de validare, raportul detaliat de validare, raportul diagnostic și raportul de validare ETSI. Pentru mai multe detalii privind raportul de validare al semnăturii vezi [29]. Rapoartele de validare conțin rezultate descrise în Tabelele 4.1 și 4.2.

5.2 Funcționalitățile serviciului de validare a semnăturii

Funcționalitățile implementate, respectiv endpoint-urile create se împart în două mari categorii: (1) funcționalități ce adresează strict verificarea certificatului, valabile pentru cazul în care un utilizator dorește să valideze un certificat fără să verifice o semnătură generată cu acesta și (2) funcționalități ce țin de verificarea semnăturii (aici desigur are loc și verificarea certificatului, dar acest proces poate fi invizibil pentru utilizator). De asemenea, trebuie subliniat faptul că aceste endpoint-uri pot fi apelate prin trimiterea unor cereri ce conțin obiecte de date de tip *form-data* sau obiecte *json* (practic au fost implementate endpoint-uri pentru ambele tipuri de date). Pentru obiectele de tip *form-data* datele semnate, datele originale, politica de validare, etc. sunt încărcate pur și simplu ca fișiere și sunt procesate la nivelul serviciului ca obiecte de tip *Multipartfile,* iar pentru obiectele de tip *json* aceste fișiere sunt codate *Base64* fiind procesate la nivelul serviciului ca obiecte de tip *String* (codarea fișierelor în formatul *Base64* a fost automatizată prin crearea unui script în limbajul *python*). Vor fi menționate doar endpoint-urile pentru obiecte *Multipartfile*, nefiind necesară și enumerarea celor pentru obiecte *json*, dat fiind faptul că răspund acelorași funcționalități.

Listă puncte de acces (endpoint-uri):

a) endpoint-urile ce țin de verificarea certificatului încep cu */certificate.*

1. */multipartfile/validation* - primește un certificat pentru verificare și întoarce un *raport simplu de validare* în format json (Anexa 2 Exemplul 1).
2. */chain/multipartfile/validation* - la fel ca la 1. În plus, utilizatorul poate adăuga și lanțul de certificate (sau doar unele certificate din lanț) pentru a fi folosite în procesul de verificare.
3. */download-xml-simple-report* - utilizatorul poate descărca raportul simplu în format XML.
4. */download-xml-detailed-report* - utilizatorul poate descărca raportul detaliat în format XML (Anexa 2 Exemplul 2).
5. */download-xml-diagnostic-data* - utilizatorul poate descărca raportul diagnostic în format XML.
6. */download-certificate* - utilizatorul (de exemplu, un alt utilizator decât cel care a încărcat certificatul pentru verificare, în contextul în care mai mulți dezvoltatori lucrează cu API-ul furnizat, iar unul dintre ei dorește să

vizualizeze un certificat încărcat de altcineva) poate să descarce certificatul verificat.

7. */download-revocation* - utilizatorul poate să descarce datele de revocare. Dacă datele sunt reprezentate de un răspuns de la server-ul de OCSP se descarcă un fișier cu extensia .ocsp. Conținutul fișierului este reprezentat în Figura 5.5. Dacă datele de revocare au fost obținute dintr-o listă de certificate revocate - CRL (server-ul de OCSP nu a putut fi accesat) se descarcă această listă în format PEM.

b) endpoint-urile ce țin de validarea semnăturii încep cu */validation*

1. */validation-form-data* - se validează semnătura pentru un singur fișier de date semnat. Se furnizează documentul original (poate fi de orice tip), documentul/fișierul semnat și opțional certificatul folosit pentru semnare (dacă se dorește descărcarea datelor de revocare). În urma procesului de validare se întoarce un raport simplu în format XML. În acest context, se validează semnături CAdES, PAdES, XAdES -B, -T, -LT, -LTA ce pot fi înfășurătoare, înfășurate, detașate. Semnăturile detașate sunt păstrate în containere de tip ASiC (ASiC-S și ASiC-E) și practic aceste containere sunt trimise procesului de validare. Pentru mai multe detalii privind ASiC vezi [30]. Procesul de validare implicit este cel de tip ARCHIVAL_DATA (4.8.2.4), acesta fiind selectat automat ca fiind procesul de validare pentru endpoint-ul curent.

2. */validation-multiple-docs-form-data* - la fel ca la 1. În plus, se poate valida o semnătură realizată peste mai multe documente (se furnizează toate documentele pentru validare). Mai mult, pentru cazul în care se validează semnături marcate temporal, se poate seta un parametru pentru *timestamp*, acest parametru reprezentând faptul că utilizatorul dorește includerea datelor temporale extrase de la nivelul semnăturii în raportul final de validare. Se mai poate alege procesul de validare dorit: *BASIC_SIGNATURES, LONG_TERM_DATA, ARCHIVAL_DATA* (vezi 4.8.2). Tot aici utilizatorul poate încărca o politică personalizată de validare (constrângerile prezentate la partea teoretică – vezi 4.7). Dacă nu se încarcă o politică în mod explicit, aplicația realizează validarea conform constrângerilor implicite existente la nivelul framework-ului.

3. */validation-add-ts-form-data* - la fel ca la 1. Prin acest endpoint se specifică în mod explicit adăugarea informațiilor de timp (*timestamp-uri*) în raportul final de validare.

4. */validation/download-diagnostic-data* - la fel ca la a) 5.

5. */validation/download-etsi-report* - utilizatorul poate descărca raportul de validare ETSI în format XML.

6. */validation/download-simple-report* - utilizatorul poate descărca raportul simplu de validare în format PDF.

7. */validation/download-detailed-report* - utilizatorul poate descărca raportul detaliat de validare în format PDF.

8. */download-certificate* - la fel ca la a) 6.

9. */download-revocation* - la fel ca la a) 7.

10. */download-timestamp* - utilizatorul poate descărca mărcile temporale extrase de la nivelul semnăturii.

c) endpoint-uri ce oferă informații cu privire la certificatele considerate de încredere:

1. */trust-certificates-info* - returnează lista de certificate încărcate din fișierul **keystore.p12** (certificatul auto-semnat *root-ca.crt* și certificatele extrase din jurnalul oficial al EU [27]) și câteva detalii aferente acestora. Figura 5.6 prezintă o înregistrare din această colecție.
2. */tsl-info* - returnează un sumar al LOTL organizat în funcție de codul țării. Practic, se afișează pentru fiecare țară câti furnizori de servicii de încredere se află în LOTL, când au fost actualizate datele la furnizorii de încredere, când va fi următoarea actualizare, etc.
3. */tsl-info/{country:[a-z][a-z]}* - returnează informații cu privire la furnizorii de servicii de încredere aflați la nivelul LOTL, în funcție de codul țării. Un exemplu în acest sens este Figura 5.7.

```
Bucharest1
ATM ROOT CA1
ocsp3@pki.com 20200220191722Z0q0o0I0 20200217005600Z 2020022019172 2Z0
Bucharest1
ATM ROOT CA1 ATM COMMON NAME1
atm_ca@pki.com0
200217133954Z
210216133954Z0w1 Bucharest1
ATM ROOT CA1
ocsp3@pki.com0
0http://localhost:8080/diverse/rootca/root-ca.crt0!† 8http://localhost:8080/diverse/rootca
/crl/crllist_der.crl0
```

Figura 5.5 Conținut fișier .ocsp


```
{
    "dssId": "C-5400AB712C41AAF0C40B505E264D5494D8AF4180F8F62955D16223B3290F97C3",
    "subjetName": "2.5.4.5=#130b363730323233333030333430,2.5.4.42=#13094a65616e2d4d617263,2.5.4.4=#13085665726265726774,CN=Jean-Marc Verbergt (Signature),C=BE",
    "issuerName": "2.5.4.5=#1306323031353038,CN=Citizen CA,C=BE",
    "notBefore": "2015-05-01T18:54:54.000+0000",
    "notAfter": "2025-04-25T23:59:59.000+0000",
    "sha1Hex": "4b 9d d1 8f 1f 2c d8 b0 5f 46 73 5d 4b 5c f8 6f 92 33 61 b4 ",
    "sha256Hex": "54 00 ab 71 2c 41 aa f0 c4 0b 50 5e 26 4d 54 94 d8 af 41 80 f8 f6 29 55 d1 62 23 b3 29 0f 97 c3 ",
    "sha1Base64": "S53Rjx8s2LBfRnNdS1z4b5IzYbQ=",
    "sha256Base64": "VACrcSxBqvDEC1BeJk1UlNivQYD49ilV0WIjsykPl8M="
},
```


Figura 5.6 Detalii certificat încărcat din keystore

```
Electronic Address; Name; PostalAddress; Registration Identifier

mailto:office@transsped.ro;  Trans Sped SRL;  38 Despot Voda Street, 020656, Bucharest, 2nd District, RO;  VATRO-12458924;

mailto:office@digisign.ro;  DigiSign S.A.;  Virgil Madgearu Street, Nr. 2-6, 014135, Bucharest, 1st District, RO;  VATRO-17544945;

http://www.certsign.ro/;  CERTSIGN S.A.;  Oltenitei Avenue, No. 107a, bl. C1, 041303, Bucharest, Sector 4, RO;  VATRO-18288250;

mailto:office@alfasign.ro;  AlfaTrust Certification S.A.;  155 Calea Victoriei, building D1, 010073, Bucharest, 1st District, RO;  VATRO-16477015;

mailto:office@certdigital.ro;  CENTRUL DE CALCUL SA;  Tudor Vladimirescu Street, No. 17, 210132, Targu Jiu, Gorj, RO;  VATRO-2163993;
```

Figura 5.7 Detalii privind TSPs din România

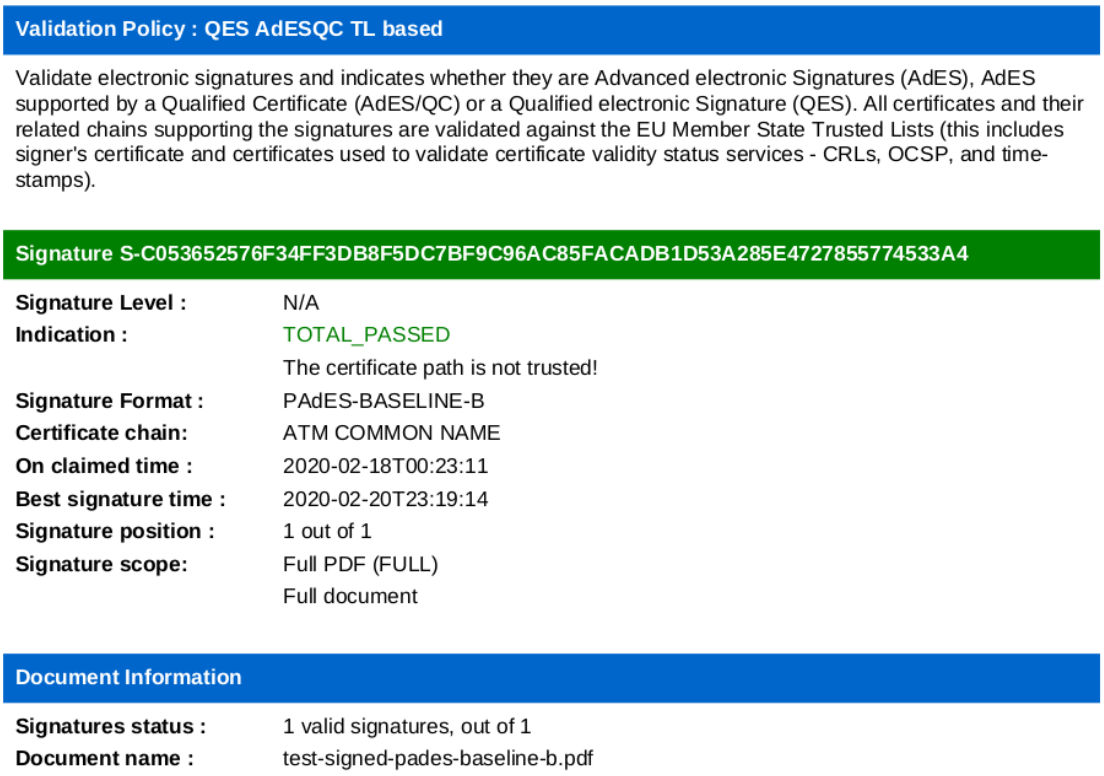
5.3 Testare și validare

Pentru crearea semnăturilor, astfel încât serviciul de validare să poată fi testat, s-au folosit [31], [32] și [33]. În urma proceselor de validare, rezultatele obținute au fost comparate cu cele prezentate în Tabelul 4.2. Astfel, dacă un certificat a fost revocat de CA, procesul de validare a întors un rezultat ce specifică acest lucru printr-un câmp prezentat în Tabelul 4.2.

Un aspect important de subliniat în această secțiune este faptul că modificarea politicii de validare poate schimba complet rezultatul procesului, după cum s-a amintit și la partea teoretică. Acest aspect s-a testat prin următorul scenariu. S-a generat un certificat de către CA-ul configurat pe server-ul tomcat. Acest certificat nu poate fi considerat de încredere (la nivelul EU) comparativ cu certificatele conținute de jurnalul oficial al EU sau de LOTL, deoarece CA-ul care l-a creat nu se află în aceste liste. Certificatul generat este considerat de încredere atât timp cât certificatul CA-ului este considerat de încredere. În acest sens, s-a creat o semnătură cu acest certificat generat, iar semnătura a fost validată în raport cu politica implicită definită de ETSI și în raport cu o politică personalizată definită pentru testare. La nivelul politicii implicite se aplică următoarele reguli:

```
<!-- eIDAS REGL 910/EU/2014 -->
<eIDAS>
    <TLFreshness Level="WARN" Unit="HOURS"
     Value="6" />
    <TLNotExpired Level="WARN" />
    <TLWellSigned Level="FAIL" />
    <TLVersion Level="FAIL" value="5" />
    <TLConsistency Level="FAIL" />
</eIDAS>
```

Aceste reguli definite de eIDAS se referă la „prospețimea”, consistența, valabilitatea, etc. listei de furnizori de încredere conținuți de LOTL. După validarea semnăturii în raport cu politica implicită (semnătura este creată de un certificat valid definit de CA, CA al cărui certificat se află în keystore-ul considerat *trusted*) rezultatul este următorul:



Validation Policy : QES AdESQC TL based

Validate electronic signatures and indicates whether they are Advanced electronic Signatures (AdES), AdES supported by a Qualified Certificate (AdES/QC) or a Qualified electronic Signature (QES). All certificates and their related chains supporting the signatures are validated against the EU Member State Trusted Lists (this includes signer's certificate and certificates used to validate certificate validity status services - CRLs, OCSP, and time-stamps).

Signature S-C053652576F34FF3DB8F5DC7BF9C96AC85FACADB1D53A285E4727855774533A4

Signature Level :	N/A
Indication :	TOTAL_PASSED
	The certificate path is not trusted!
Signature Format :	PAdES-BASELINE-B
Certificate chain:	ATM COMMON NAME
On claimed time :	2020-02-18T00:23:11
Best signature time :	2020-02-20T23:19:14
Signature position :	1 out of 1
Signature scope:	Full PDF (FULL)
	Full document

Document Information

Signatures status :	1 valid signatures, out of 1
Document name :	test-signed-pades-baseline-b.pdf

Figura 5.8 Rezultat validare politică implicită. (raport simplu în format PDF)

Astfel, se poate observa că semnătura a trecut cu succes toate procesele de validare aplicate, procese descrise la partea teoretică, și că lanțul de certificate, ce se termină cu CA-ul creat, nu este considerat de încredere. În Figura 5.8 se poate observa și descrierea politicii de validare folosite.

În cazul folosirii politicii personalizate (au fost eliminate regulile eIDAS) rezultatul este următorul:

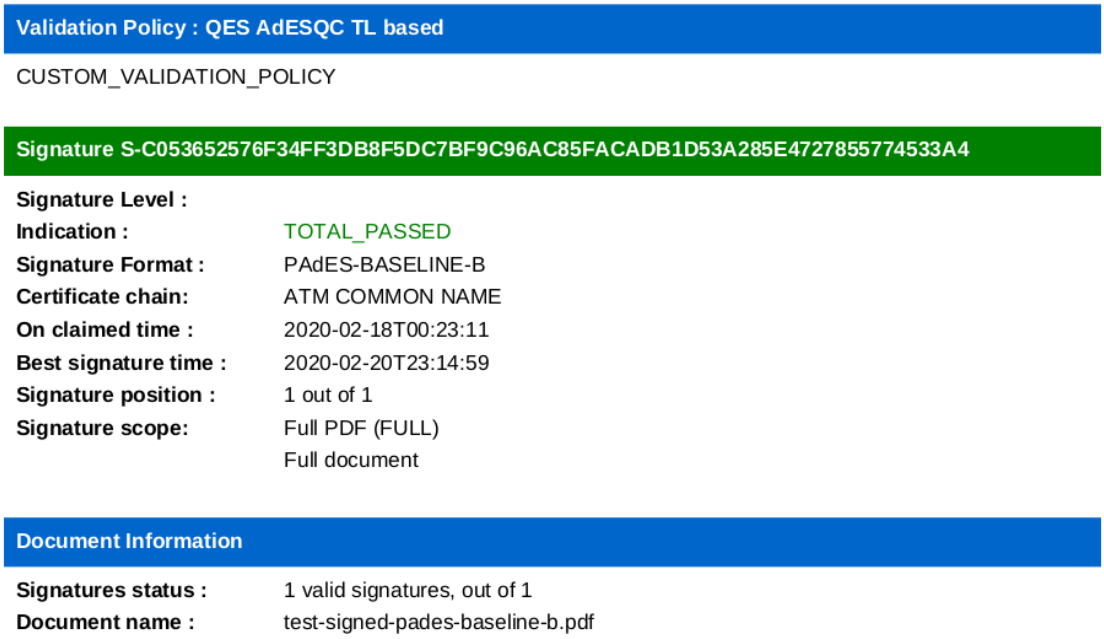


Figura 5.9 Rezultat validare politică definită de utilizator (raport simplu în format PDF)

În Figura 5.9 se observă că toate procesele de validare s-au finalizat cu succes și că lanțul de certificare este considerat de încredere.

Exemplul prezentat este unul minimal și destul de simplu, dar un utilizator trebuie să fie foarte atent atunci când decide să utilizeze o politică de validare personalizată. Acesta poate modifica toate constrângerile de validare (constrângeri de validare X.509, constrângeri criptografice și constrângeri referitoare la elementele semnăturii) putând schimba în mod categoric rezultatul unui proces de validare.

Pentru testarea nivelului de calificare al semnăturii s-au folosit testele de la [37].

Rezultatele obținute sunt prezentate în Tabelul 5.1

Tabel 5.1 Rezutate teste

Fișier test	Rezultat Așteptat	Rezultat Obținut
2.1.1-TEST-FILE.xml	AdESig	AdESig
2.1.2-TEST-FILE.xml	AdESig	AdESig
2.1.3-TEST-FILE.xml	QESig	QESig
7.4.1-TEST-FILE.xml	QESig	QESig
7.1.1-TEST-FILE.xml	AdESig	AdESig
5.1.3-TEST-FILE.xml	QESig	QESig
4.2.3-TEST-FILE.xml	AdESig-QC	AdESig-QC
6.4.6-TEST-FILE.xml	N/A	N/A
6.3.2-TEST-FILE.xml	AdESig-QC	AdESig-QC
4.3.6-TEST-FILE.xml	AdESeal	AdESeal

Concluzii

În contextul societății actuale ce se bazează în aproape orice tip de activitate pe tranzacții electronice, devine obligatorie validarea acestor tranzacții. În acest sens, furnizorii de servicii de încredere (TSP) au o mare responsabilitate în ceea ce privește execuția și validarea tranzacțiilor electronice. Aceste tranzacții au la bază criptosisteme cu chei publice și mai exact semnături digitale. Având în vedere acest aspect, rezultă că semnăturile digitale în sine trebuie să treacă printr-un proces de verificare. Astfel, teza de față prezintă un sistem creat pentru verificarea semnăturilor și a certificatelor digitale. Se pune accentul pe un serviciu de validare a semnăturii de tip REST. Serviciul este dezvoltat folosindu-se limbajul de prgramare Java și mai exact, framework-ul Spring Boot. De asemenea, teza conține și un memoriu tehnic pentru înțelegerea contextului și a modului în care un proces de validare are loc. Pe tot parcursul lucrării s-au făcut referiri la specificațiile tehnice definite de ETSI, acestea reprezentând baza sistemului dezvoltat.

Bibliografie

1. S.R. Subramanya, Digital signatures. IEEE Potentials. April 2006.
2. M. Togan și I. Florea, Infrastructuri de securitate pentru servicii electornice în Internet, Matrix Rom, 2017.
3. European Union, Regulamentul nr. 910/2014 AL PARLAMENTULUI EUROPEAN ȘI AL CONSILIULUI din 23 iulie 2014 privind identificarea electronică și serviciile de încredere pentru tranzacțiile electronice pe piața internă și de abrogare a Directivei 1993/93/CE, 2014.
4. IETF RFC 5652, https://tools.ietf.org/html/rfc5652
5. ETSI TS 101 733 v2.2.1 (2013-04), Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signature (CAdES), https://www.etsi.org/deliver/etsi_ts/101700_101799/101733/02.02.01_60/ts_101733v020201p.pdf
6. ETSI TS 101 903 v1.4.1 (2009-06), XML Advanced Electronic Signatures (XAdES) https://www.etsi.org/deliver/etsi_ts/101900_101999/101903/01.04.01_60/ts_101903v010401p.pdf
7. ETSI TS 102 778 v1.1.1 (2009-07), Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles; Part 1:PAdES Overview - a framework document for PAdES, https://www.etsi.org/deliver/etsi_ts/102700_102799/10277801/01.01.01_60/ts_10277801v010101p.pdf
8. M. Bartel, J. Boyer, B. Fox, B. LaMacchia, E. Simon. XML Signature Syntax and Processing (Second Edition). W3C Recommendation, 2008.
9. M. Bartel, J. Boyer, B. Fox, B. LaMacchia, E. Simon. XML Signature Syntax and Processing Verison 2.0, W3C Working Group Note 23 July 2015, https://www.w3.org/TR/xmldsig-core2/
10. ETSI TS 119 441 v1.1.1 (2018-08), Electronic Signatures and Infrastructures (ESI); Policy requirements for TSP providing signature validation services. https://www.etsi.org/deliver/etsi_ts/119400_119499/119441/01.01.01_60/ts_119441v010101p.pdf
11. List of Trusted Lists or List of the Lists - LOTL, https://ec.europa.eu/tools/lotl/eu-lotl.xml

12. ETSI TS 119 102-1 v1.2.1 (2018-08), Electronic Signatures and Infrastructures (ESI); Procedures for Creation and Validation of AdES Digital Signatures; Part 1: Creation and Validation, https://www.etsi.org/deliver/etsi_ts/119100_119199/11910201/01.02.01_60/ts_119102 01v010201p.pdf

13. ETSI TS 119 172-1 v1.1.1 (2015-07), Electronic Signatures and Infrastructures (ESI); Signature Policies; Part 1: Building blocks and table of contents for human readable signature policy documents, https://www.etsi.org/deliver/etsi_ts/119100_119199/11917201/01.01.01_60/ts_119172 01v010101p.pdf

14. IETF RFC 3161, https://www.ietf.org/rfc/rfc3161.txt

15. ETSI EN 319 442 v1.1.0 (2015-12), Electronic Signatures and Infrastructures (ESI); Time-stamping protocol and time-stamp token profiles, https://www.etsi.org/deliver/etsi_en/319400_319499/319422/01.01.00_30/en_319422v010100v.pdf .

16. Digital Signature Service version: 5.5-2019-10-15, https://ec.europa.eu/cefdigital/DSS/webapp-demo/doc/dss-documentation.pdf

17. Digital Signature Services (DSS), DSS releases, https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/DSS+releases

18. Spring Boot - Introduction, https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm?fbclid=IwAR25okNoqUTtPdCqoafJ8u9F_qtwgb-KCZ83WAmS3sh3IolYiXVUsqgnXok

19. NginX, https://en.wikipedia.org/wiki/Nginx?fbclid=IwAR3pEcpVfa6z_rC0F2dkZ62MqaJiXgNW5KrWV3WGhvigXf7pXBvDzVny_Lw

20. Reverse proxy, https://en.wikipedia.org/wiki/Reverse_proxy

21. Postman, https://www.getpostman.com/

22. OpenSSL, https://www.openssl.org/

23. IETF RFC 6960, https://tools.ietf.org/html/rfc6960

24. HikariCP, https://github.com/brettwooldridge/HikariCP

25. HyperSQL, http://hsqldb.org/

26. HSQLDB Tutorial, https://www.tutorialspoint.com/hsqldb/index.htm

27. Official Journal Of the Union, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.C_.2019.276.01.0001.01.ENG

28. LOTL, https://ec.europa.eu/tools/lotl/eu-lotl.xml.

29. ETSI TS 119 102-2 v1.1.1 (2018-08), Electronic Signatures and Infrastructures (ESI); Procedures for Creation and Validation of AdES Digital Signatures; Part 2. Signature Validation Report. https://www.etsi.org/deliver/etsi_ts/119100_119199/11910202/01.01.01_60/ts_111910202v010101p.pdf

30. ETSI TS 102 918 v1.1.1 (2011-04), Electronic Signatures and Infrastructures (ESI); Associated Signature Containers (ASiC), https://www.etsi.org/deliver/etsi_ts/102900_102999/102918/01.01.01_60/ts_102918v010101p.pdf

31. Sign a document, https://ec.europa.eu/cefdigital/DSS/webapp-demo/sign-a-document

32. Sign a PDF,

https://ec.europa.eu/cefdigital/DSS/webapp-demo/sign-a-pdf

33. Sign multiple documents, https://ec.europa.eu/cefdigital/DSS/webapp-demo/sign-multiple-documents

34. ETSI (European Telecommunications Standards Institute). Gartner.Retrieved 27 September 2018. https://www.gartner.com/it-glossary/etsi-european-telecommunications-standards-institute

35. ETSI. https://en.wikipedia.org/wiki/ETSI

36. European Commission, Points of Single Contact, https://ec.europa.eu/growth/single-market/services/services-directive/in-practice/contact_en

37. Test cases packages, https://webgate.ec.europa.eu/esig-validation-tests/testcases

38. https://github.com/StefanBodoarca/disertatie

Anexa 1

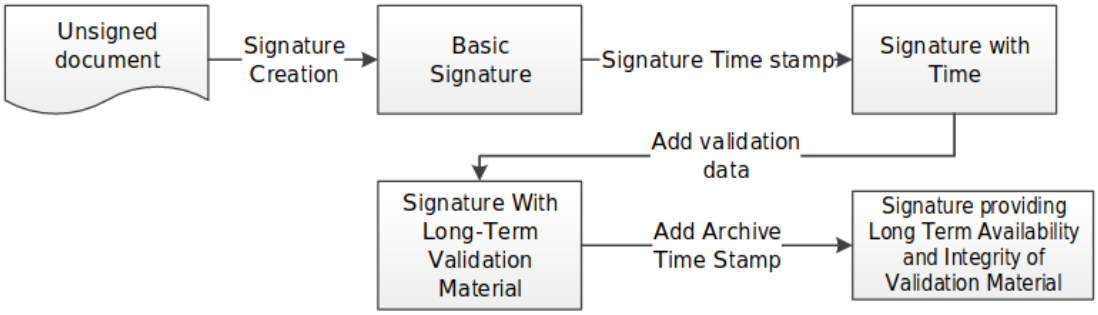


Figura 1 Ciclul de viață al semnăturii [12]

Figura 2 Semnătură de bază [12]

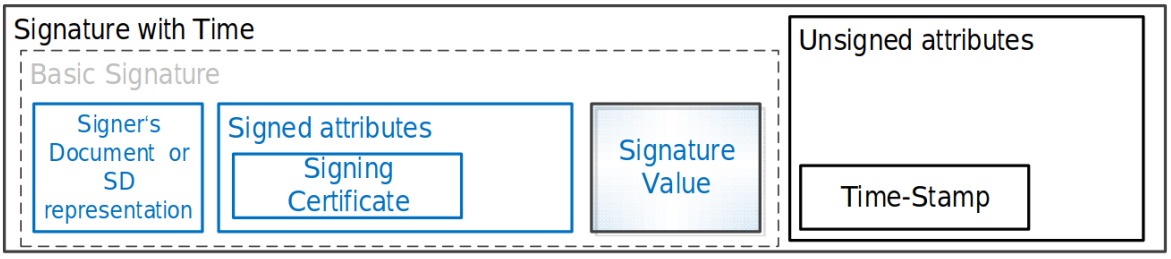


Figura 3 Semnătură cu marcă temporală [12]

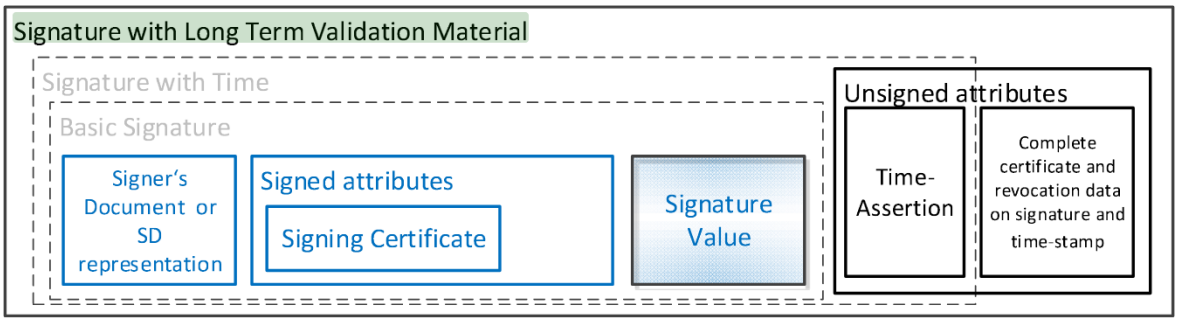


Figura 4 SignatureLTVM [12] (Def. 3)

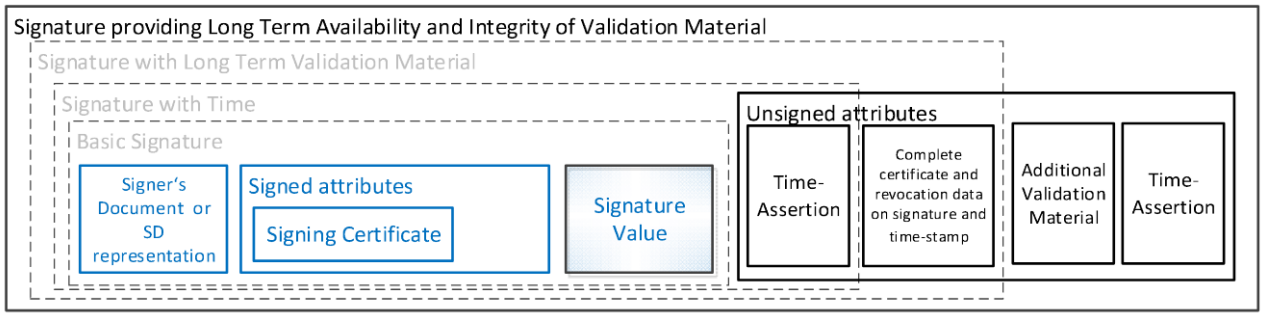


Figura 5 SignatureLTAIVM [12] (Def 4.)

Anexa 2

Exemplul 1 Raport simplu rezultat în urma validării certificatului


```
{
  "jaxbModel": {
    "chain": [
      {
        "id": "C-49FED92E01DEDDEE29733531470BC724EE9B7EA7A79742C75E29290F1B0F8806",
        "subject": {
          "commonName": "x6",
          "surname": null,
          "givenName": null,
          "pseudonym": null,
          "organizationName": "ATM ORG",
          "organizationUnit": "ATM ROOT CA",
          "email": "x6@pki.com",
          "locality": null,
          "state": "Bucharest",
          "country": "RO"
        },
        "issuerId": "C-
9FA2E3CDC792930DC9B6EB8E3A8AEE5D78DCD5AAEA007B80EBB927D817E32758",
        "notBefore": "2020-02-08T20:40:47.000+0000",
        "notAfter": "2021-02-07T20:40:47.000+0000",
        "keyUsages": [
          "DIGITAL_SIGNATURE",
          "NON_REPUDIATION",
          "KEY_ENCIPHERMENT",
          "DATA_ENCIPHERMENT",
          "KEY_AGREEMENT"
```

```
    ],
    "extendedKeyUsages": [
        "clientAuth",
        "serverAuth"
    ],
    "ocspUrls": [
        "http://127.0.0.1:8888"
    ],
    "crlUrls": [
        "http://localhost:8080/diverse/rootca/crl/crllist_der.crl"
    ],
    "aiaUrls": [
        "http://localhost:8080/diverse/rootca/root-ca.crt"
    ],
    "cpsUrls": null,
    "pdsUrls": null,
    "qualificationAtIssuance": "NA",
    "qualificationAtValidation": "NA",
    "revocation": {
        "productionDate": "2020-02-17T00:56:42.000+0000",
        "revocationDate": "2020-02-17T00:56:00.000+0000",
        "revocationReason": null
    },
    "trustAnchors": null,
    "indication": "INDETERMINATE",
    "subIndication": "REVOKED_NO_POE"
},
{
    "id": "C-9FA2E3CDC792930DC9B6EB8E3A8AEE5D78DCD5AAEA007B80EBB927D817E32758",
```

```
"subject": {
    "commonName": "ATM COMMON NAME",
    "surname": null,
    "givenName": null,
    "pseudonym": null,
    "organizationName": "ATM ORG",
    "organizationUnit": "ATM ROOT CA",
    "email": "atm_ca@pki.com",
    "locality": null,
    "state": "Bucharest",
    "country": "RO"
},
"issuerId": null,
"notBefore": "2020-02-08T17:47:15.000+0000",
"notAfter": "2021-02-07T17:47:15.000+0000",
"keyUsages": [
    "KEY_CERT_SIGN",
    "CRL_SIGN"
],
"extendedKeyUsages": null,
"ocspUrls": null,
"crlUrls": null,
"aiaUrls": null,
"cpsUrls": null,
"pdsUrls": null,
"qualificationAtIssuance": null,
"qualificationAtValidation": null,
"revocation": {
    "productionDate": null,
```

```
            "revocationDate": null,
            "revocationReason": null
          },
          "trustAnchors": [],
          "indication": "PASSED",
          "subIndication": null
        }
      ],
      "validationTime": "2020-02-20T18:50:46.943+0000"
    },
    "validationTime": "2020-02-20T18:50:46.943+0000",
    "certificateIds": [
      "C-49FED92E01DEDDEE29733531470BC724EE9B7EA7A79742C75E29290F1B0F8806",
      "C-9FA2E3CDC792930DC9B6EB8E3A8AEE5D78DCD5AAEA007B80EBB927D817E32758"
    ],
    "qualificationAtCertificateIssuance": "NA",
    "trustAnchorVATNumbers": [],
    "qualificationAtValidationTime": "NA"
}
```


Exemplul 2 Raport xml detaliat rezultat în urma validării certificatului


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<DetailedReport xmlns="http://dss.esig.europa.eu/validation/detailed-report">
  <Certificate Title="Certificate Qualification">
    <Constraint>
      <Name NameId="BBB_ACCEPT">Is the result of the Basic Building Block acceptable?</Name>
      <Status>WARNING</Status>
```

```
        <Warning NameId="BBB_ACCEPT_ANS">The result of the Basic Building Block is not acceptable!</Warn
ing>
      </Constraint>
      <Conclusion>
        <Indication>INDETERMINATE</Indication>
        <Warnings NameId="BBB_ACCEPT_ANS">The result of the Basic Building Block is not acceptable!</War
nings>
      </Conclusion>
    </Certificate>
    <BasicBuildingBlocks Id="C-
49FED92E01DEDDEE29733531470BC724EE9B7EA7A79742C75E29290F1B0F8806" Type="CERTIFICATE">
      <XCV Title="X509 Certificate Validation">
        <Constraint>
          <Name NameId="BBB_XCV_CCCBB">Can the certificate chain be built till the trust anchor?</Name>
          <Status>OK</Status>
        </Constraint>
        <Constraint Id="C-
49FED92E01DEDDEE29733531470BC724EE9B7EA7A79742C75E29290F1B0F8806">
          <Name NameId="BBB_XCV_SUB">Is the certificate validation concluant ?</Name>
          <Status>NOT OK</Status>
          <Error NameId="BBB_XCV_SUB_ANS">The certificate validation is not concluant!</Error>
        </Constraint>
        <Conclusion>
          <Indication>INDETERMINATE</Indication>
          <SubIndication>REVOKED_NO_POE</SubIndication>
          <Errors NameId="BBB_XCV_ISCR_ANS">The certificate is revoked!</Errors>
        </Conclusion>
        <SubXCV Id="C-
49FED92E01DEDDEE29733531470BC724EE9B7EA7A79742C75E29290F1B0F8806" TrustAnchor="false" Title
="Certificate Id = C-49FED92E01DEDDEE29733531470BC724EE9B7EA7A79742C75E29290F1B0F8806">
          <Constraint>
```

<Name NameId="QUAL_UNIQUE_CERT">Is the certificate unique ?</Name>

<Status>OK</Status>

</Constraint>

<Constraint>

<Name NameId="BBB_XCV_PSEUDO_USE">Is pseudo used ?</Name>

<Status>OK</Status>

</Constraint>

<Constraint>

<Name NameId="BBB_XCV_ISNSSC">Is not self-signed certificate?</Name>

<Status>OK</Status>

</Constraint>

<Constraint>

<Name NameId="BBB_XCV_ICSI">Is the certificate's signature intact?</Name>

<Status>OK</Status>

</Constraint>

<Constraint>

<Name NameId="ACCCM">Are certificate cryptographic constraints met?</Name>

<Status>OK</Status>

<AdditionalInfo>Validation time : 2020-02-20 18:50 for token with ID : [C-49FED92E01DEDDEE29733531470BC724EE9B7EA7A79742C75E29290F1B0F8806]</AdditionalInfo>

</Constraint>

<Constraint>

<Name NameId="BBB_XCV_ISCGKU">Has the signer's certificate given key-usage?</Name>

<Status>OK</Status>

<AdditionalInfo>Key usage : [DIGITAL_SIGNATURE, NON_REPUDIATION, KEY_ENCIPHERMENT, DATA_ENCIPHERMENT, KEY_AGREEMENT]</AdditionalInfo>

</Constraint>

<Constraint>

<Name NameId="BBB_XCV_AIA_PRES">Is authority info access present?</Name>

<Status>OK</Status>

```
            </Constraint>
            <Constraint>
               <Name NameId="BBB_XCV_REVOC_PRES">Is revocation info access present?</Name>
               <Status>OK</Status>
            </Constraint>
            <Constraint>
               <Name NameId="BBB_XCV_ICTIVRSC">Is the current time in the validity range of the signer's certificate?</Name>
               <Status>OK</Status>
               <AdditionalInfo>Certificate validity : 2020-02-08 20:40 to 2021-02-07 20:40</AdditionalInfo>
            </Constraint>
            <Constraint>
               <Name NameId="BBB_XCV_ISCR">Is the certificate not revoked?</Name>
               <Status>NOT OK</Status>
               <Error NameId="BBB_XCV_ISCR_ANS">The certificate is revoked!</Error>
               <AdditionalInfo>Revocation reason : null (date : 2020-02-17 00:56)</AdditionalInfo>
            </Constraint>
            <Conclusion>
               <Indication>INDETERMINATE</Indication>
               <SubIndication>REVOKED_NO_POE</SubIndication>
               <Errors NameId="BBB_XCV_ISCR_ANS">The certificate is revoked!</Errors>
            </Conclusion>
            <RFC Id="R-86E9D43B86547AE2902D86738666E505CA8431D9F978B1584400C57FC1D1690B" Title="Revocation Freshness Checker">
               <Constraint>
                  <Name NameId="BBB_XCV_IRDPFC">Is the revocation data present for the certificate?</Name>
                  <Status>OK</Status>
               </Constraint>
               <Constraint>
```

```
                <Name NameId="BBB_RFC_NUP">Is there a Next Update defined for the revocation data?</Nam
e>
                <Status>OK</Status>
              </Constraint>
              <Constraint>
                <Name NameId="BBB_RFC_IRIF">Is the revocation information fresh for the certificate?</Name>
                <Status>IGNORED</Status>
              </Constraint>
              <Constraint>
                <Name NameId="ARCCM">Are revocation cryptographic constraints met?</Name>
                <Status>OK</Status>
                <AdditionalInfo>Validation time : 2020-02-20 18:50 for token with ID : [R-
86E9D43B86547AE2902D86738666E505CA8431D9F978B1584400C57FC1D1690B]</AdditionalInfo>
              </Constraint>
              <Conclusion>
                <Indication>PASSED</Indication>
              </Conclusion>
            </RFC>
            <RevocationInfo>
              <CertificateId>C-
49FED92E01DEDDEE29733531470BC724EE9B7EA7A79742C75E29290F1B0F8806</CertificateId>
              <RevocationId>R-
86E9D43B86547AE2902D86738666E505CA8431D9F978B1584400C57FC1D1690B</RevocationId>
              <RevocationDate>2020-02-17T00:56:00</RevocationDate>
            </RevocationInfo>
          </SubXCV>
          <SubXCV Id="C-
9FA2E3CDC792930DC9B6EB8E3A8AEE5D78DCD5AAEA007B80EBB927D817E32758" TrustAnchor="true" Titl
e="Certificate Id = C-9FA2E3CDC792930DC9B6EB8E3A8AEE5D78DCD5AAEA007B80EBB927D817E32758">
            <Conclusion>
              <Indication>PASSED</Indication>
```

```
            </Conclusion>
          </SubXCV>
        </XCV>
        <Conclusion>
          <Indication>INDETERMINATE</Indication>
          <SubIndication>REVOKED_NO_POE</SubIndication>
          <Errors NameId="BBB_XCV_ISCR_ANS">The certificate is revoked!</Errors>
        </Conclusion>
      </BasicBuildingBlocks>
    </DetailedReport>
```